Delhi University

B.Sc. (Computer Science)

ML Project

Submitted By :- Divya Bhardwaj (20191412)
Tannu Yadav (20191443)
Harpreet Kumari (20191460)

Submitted To :- Mr. Nikhil

Credit Card Fraud detection with Machine Learning

Problem Statement

In this, we are going to detect credit card fraud. How correctly the detection can be performed.

Data

This data is downloaded from kaggle :- https://www.kaggle.com/code/indikab/credit-card-fraud-detection-with-99-9-accuracy/data

This is a classification problem that's why these models are selected.

The models used in this are

Logistic Regression -Logistic regression, despite its name, is a classification model rather than regression model. Logistic regression is a simple and more efficient method for binary and linear classification problems. It is a classification model, which is very easy to realize and achieves very good performance with linearly separable classes. It is an extensively employed algorithm for classification in industry.

Logistic Function -$1/(1+e^{-x})$

Types of Logistic Regression

1. Binary Logistic Regression

The categorical response has only two 2 possible outcomes. Example: Spam or Not

2. Multinomial Logistic Regression

Three or more categories without ordering. Example: Predicting which food is preferred more (Veg, Non-Veg, Vegan)

3. Ordinal Logistic Regression

Three or more categories with ordering. Example: Movie rating from 1 to 5

Bernoulli Naïve Bayes - **Bernoulli Naive Bayes** is a variant of Naive Bayes.This is used for discrete data and it works on Bernoulli distribution. The main feature of Bernoulli Naive Bayes is that it accepts features only as binary values like true or false, yes or no, success or failure, 0 or 1 and so on. So when the feature values are binary we know that we have to use Bernoulli Naive Bayes classifier. **Bernoulli Naive Bayes Classifier is based on the following rule:**

$$P(x_i \mid y) = P(i \mid y)\, x_i + (1 - P(i \mid y))(1 - x_i)$$

Advantages of Bernoulli Naive Bayes:

1. 1.They are extremely fast as compared to other classification models
2.
3. 2.As in Bernoulli Naive Bayes each feature is treated independently with binary values only, it explicitly gives penalty to the model for non-occurrence of any of the features which are necessary for predicting the output y. And the other multinomial variant of Naive Bayes ignores this features instead of penalizing.
4. 3.In case of small amount of data or small documents (for example in text classification), Bernoulli Naive Bayes gives more accurate and precise results as compared to other models.

5. 4.It is fast and is able to make to make real-time predictions
6. 5.It can handle irrelevant features nicely
7. 6.Results are self-explanatory.

Random Forest Classifier + gini - A random forest is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting. The sub-sample size is controlled with the max_samples parameter if bootstrap=True (default), otherwise the whole dataset is used to build each tree. Random forest, like its name implies, consists of a large number of individual decision trees that operate as an ensemble. Each individual tree in the random forest spits out a class prediction and the class with the most votes becomes our model's prediction.

The fundamental concept behind random forest is a simple but powerful one — the wisdom of crowds. In data science speak, the reason that the random forest model works so well is:

A large number of relatively uncorrelated models (trees) operating as a committee will outperform any of the individual constituent models.

a) Gini : - The internal working of Gini impurity is also somewhat similar to the working of entropy in the decision tress. In the Decision tree algorithm, both are used for building the tree by splitting as per the appropriate features but there is quite a difference in the computation of both the methods. The **gini impurity** is calculated using the following formula:

$$\text{GiniIndex} = 1 - \sum j p_j^2$$

Where p_j is the probability of class j.

The gini impurity measures the frequency at which any element of the dataset will be mislabelled when it is randomly labeled.

The minimum value of the Gini Index is 0. This happens when the node is **pure**, this means that all the contained elements in the node are of one unique class. Therefore, this node will not be split again. Thus, the optimum split is chosen by the features with less Gini Index. Moreover, it gets the maximum value when the probability of the two classes are the same.

$$Gini_{min}=1–(1^2)=0$$

$$Gini_{max}=1–(0.5^2+0.5^2)=0.5$$

b) Entropy :- Entropy helps us to build an appropriate decision tree for selecting the best splitter. Entropy can be defined as a measure of the purity of the sub split. Entropy always lies between 0 to 1. The **entropy** is calculated using the following formula:

$$Entropy=–\sum jp_j\cdot\log_2\cdot p_j$$

Where, as before, p_j is the probability of class j.

Entropy is a measure of information that indicates the disorder of the features with the target. Similar to the Gini Index, the optimum split is chosen by the feature with less entropy. It gets its maximum value when the probability of the two classes is the same and a node is pure when the entropy has its minimum value, which is 0:

$$Entropy_{min}=−1\cdot\log_2(1)=0$$

$$Entropy_{max}=–0.5\cdot\log_2(0.5)–0.5\cdot\log_2(0.5)=1$$

Gradient Boosting Classifier :- It involves building a strong model by using a collection (or "ensemble") of "weaker" models. Gradient boosting falls under the category of boosting methods, which iteratively learn from each of the weak learners to build a strong model. It can optimize:

- Regression
- Classification
- Ranking

Gradient boosting has found applications across various technical fields. The algorithm can look complicated at first, but in most cases we use only one predefined configuration for classification and one for regression, which can of course be modified based on your requirements. In this article we'll focus on Gradient Boosting for classification problems.

Gradient Boosting has three main components:

Loss Function - The role of the loss function is to estimate how good the model is at making predictions with the given data. This could vary depending on the problem at hand. For example, if we're trying to predict the weight of a person depending on some input variables (a regression problem), then the loss function would be something that helps us find the difference between the predicted weights and the observed weights. On the other hand, if we're trying to categorize if a person will like a certain movie based on their personality, we'll require a loss function that helps us understand how accurate our model is at classifying people who did or didn't like certain movies.

Weak Learner - A weak learner is one that classifies our data but does so poorly, perhaps no better than random guessing. In other words, it has

a high error rate. These are typically decision trees (also called decision stumps, because they are less complicated than typical decision trees).

Additive Model - This is the iterative and sequential approach of adding the trees (weak learners) one step at a time. After each iteration, we need to be closer to our final model. In other words, each iteration should reduce the value of our loss function.

The best model for this data which gives us the highest accuracy is Random Forest Classifier.