Credit Card Fraud Detection

Abstract:

With many research works focusing on tackle frauds of credit card transaction or insurance, only few mentioned the identity fraud of credit card application. So, in this project we will be building a model with few machine learning models to detect such fraud.

Introduction:

When your card is lost or stolen, an unauthorized charge can happen; in other words, the person who finds it uses it for a purchase. Criminals can also forge your name and use the card or order some goods through a mobile phone or computer. Also, there is the problem of using a counterfeit credit card – a fake card that has the real account information that was stolen from holders. That is especially dangerous because the victims have their real cards, but do not know that someone has copied their card. Such fraudulent cards look quite legitimate and have the logos and encoded magnetic strips of the original one. Fraudulent credit cards are usually destroyed by the criminals after several successful payments, just before a victim realizes the problem and reports it.

The traditional approach to identify frauds is expert system, which is a set of rules made by experts, and will determine whether a transaction is fraudulent or not. However, as financial systems get more and more complicated, the number and complexity of rules grow to a point where no one could construct and maintain such complex system. As a result, more and more attention has been focused on machine learning and data mining.

Instead of writing rules by hand, computers can learn the patterns and signals of fraudulent activities and identify potential frauds based on some relatively simple algorithms. And with the development of more powerful and tailor-made hardware for training such models, handling huge data sets containing billions of records became plausible. Researchers have been building such models with a wide range of algorithms and achieved excellent accuracy. Most research, though, focused on credit card payment or transactions, but the area of identity theft, especially for credit card application, remains open. In this project, we investigate the performance and possibility of machine learning algorithms to detect fraudulent credit card applications.

Related Work:

Logistic Regression:

Logistic regression is a statistical model that in its basic form uses a logistic function to model a binary dependent variable, although many more complex extensions exist. In regression analysis, logistic regression is estimating the parameters of a logistic model (a form of binary regression). Mathematically, a binary logistic model has a dependent variable with two possible values, such as pass/fail which is represented by an indicator variable, where the two values are labeled "0" and "1". In the logistic model, the log-odds (the logarithm of the odds) for the value labeled "1" is a linear combination of one or more independent variables ("predictors"); the independent variables can each be a binary variable (two classes, coded by an indicator variable) or a continuous variable (any real value). The corresponding probability of the value labeled "1" can vary between 0 (certainly the value "0") and 1 (certainly the value "1"), hence the labeling; the function that converts log-odds to probability is the logistic function, hence the name. The unit of measurement for the log-odds scale is called a logit, from logistic unit, hence the alternative names. Analogous models with a different sigmoid function instead of the logistic function can also be used, such as the probit model; the defining characteristic of the logistic model is that increasing one of the independent variables multiplicatively scales the odds of the given outcome at a constant rate, with each independent variable having its own parameter; for a binary dependent variable this generalizes the odds ratio.

Support Vector Machines:

The objective of the support vector machine algorithm is to find a hyperplane in an N-dimensional space(N — the number of features) that distinctly classifies the data points.

To separate the two classes of data points, there are many possible hyperplanes that could be chosen. Our objective is to find a plane that has the maximum margin, i.e the maximum distance between data points of both classes. Maximizing the margin distance provides some reinforcement so that future data points can be classified with more confidence.

Hyperplanes and Support Vectors

Hyperplanes are decision boundaries that help classify the data points. Data points falling on either side of the hyperplane can be attributed to different classes. Also, the dimension of the hyperplane depends upon the number of features. If the number of input features is 2, then the hyperplane is just a line. If the number of input features is 3, then the hyperplane becomes a two-dimensional plane. It becomes difficult to imagine when the number of features exceeds 3.

Support vectors are data points that are closer to the hyperplane and influence the position and orientation of the hyperplane. Using these support vectors, we maximize the margin of the classifier. Deleting the support vectors will change the position of the hyperplane. These are the points that help us build our SVM.

Naïve bayer's:

Naive Bayes is a simple technique for constructing classifiers: models that assign class labels to problem instances, represented as vectors of feature values, where the class labels are drawn from some finite set. There is not a single algorithm for training such classifiers, but a family of algorithms based on a common principle: all naive Bayes classifiers assume that the value of a particular feature is independent of the value of any other feature, given the class variable. For example, a fruit may be considered to be an apple if it is red, round, and about 10 cm in diameter. A naive Bayes classifier considers each of these features to contribute independently to the probability that this fruit is an apple, regardless of any possible correlations between the color, roundness, and diameter features.

For some types of probability models, naive Bayes classifiers can be trained very efficiently in a supervised learning setting. In many practical applications, parameter estimation for naive Bayes models uses the method of maximum likelihood; in other words, one can work with the naive Bayes model without accepting Bayesian probability or using any Bayesian methods.

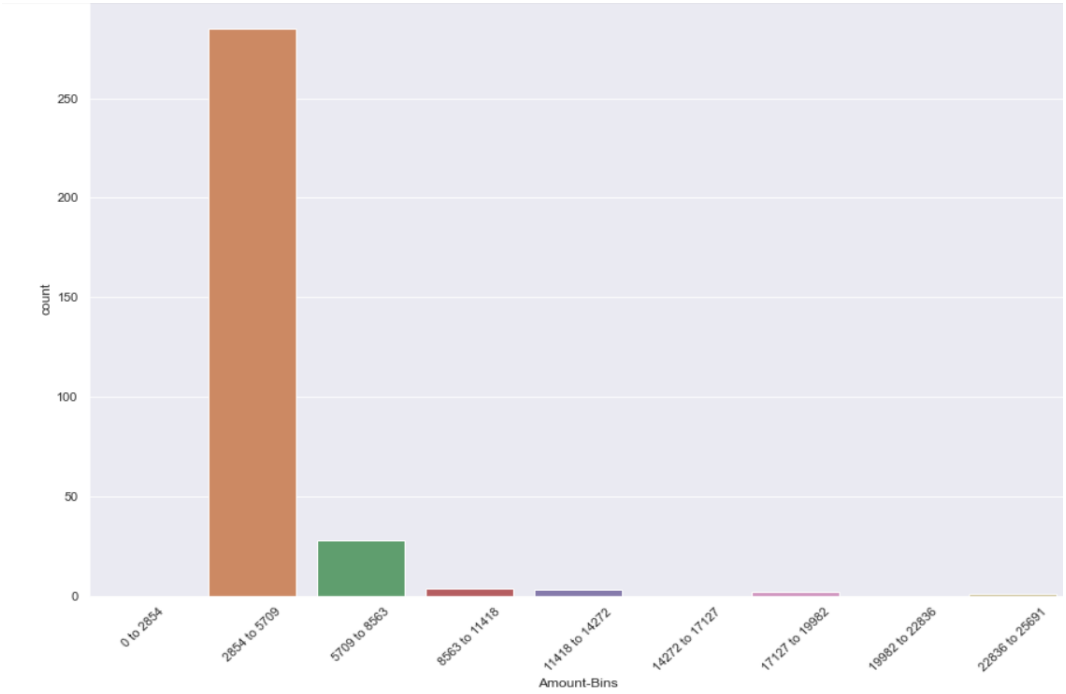
Despite their naive design and apparently oversimplified assumptions, naive Bayes classifiers have worked quite well in many complex real-world situations. In 2004, an analysis of the Bayesian classification problem showed that there are sound theoretical reasons for the apparently implausible efficacy of naive Bayes classifiers.] Still, a comprehensive comparison with other classification algorithms in 2006 showed that Bayes classification is outperformed by other approaches, such as boosted trees or random forests.

An advantage of naive Bayes is that it only requires a small number of training data to estimate the parameters necessary for classification.

Methodology:

First, we performed EDA to understand our data. The columns V1-28 are the result of PCA dimensional reduction so as to protect user identity and sensitive features. We then saw the distribution of the column ‘Amount’ in our dataset and found its highly imbalanced.

So we used binning to classify them into different categories so as to learn more about it.

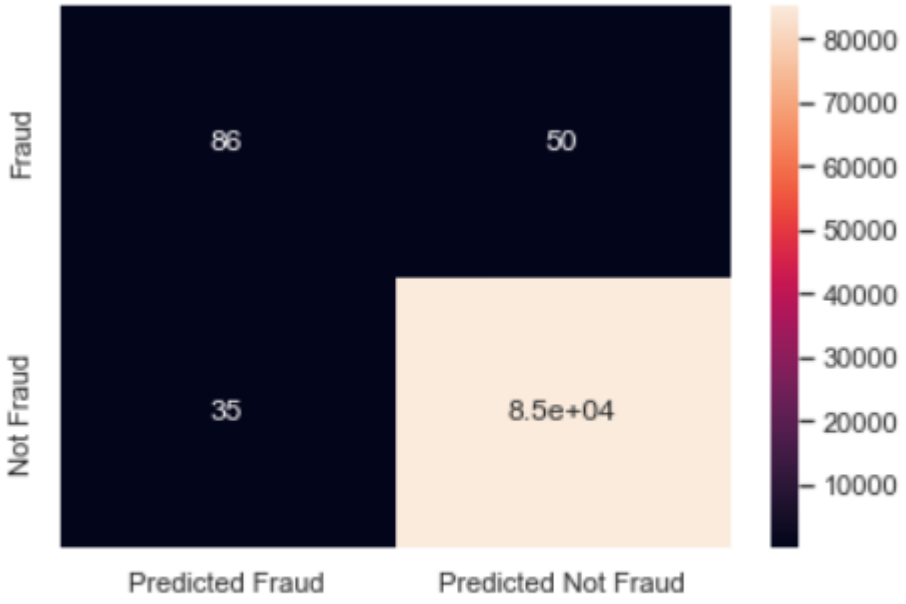


After understanding our data and making it ready for using it in our models, we split our data into training set and testing set. We fit out models with the training data and later tested it with the test set.

Some of the results obtained are:

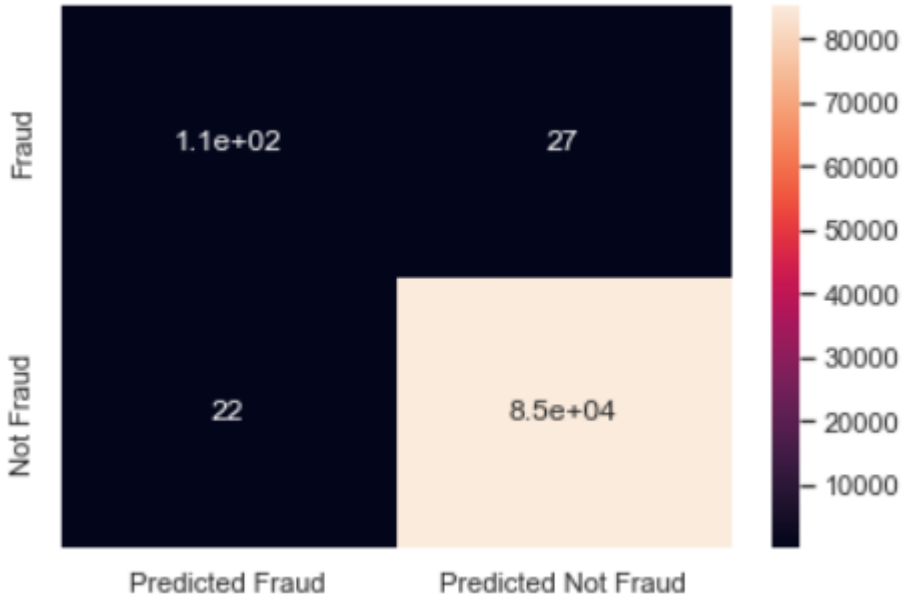
Results:

Confusion matrix for logistic regression.

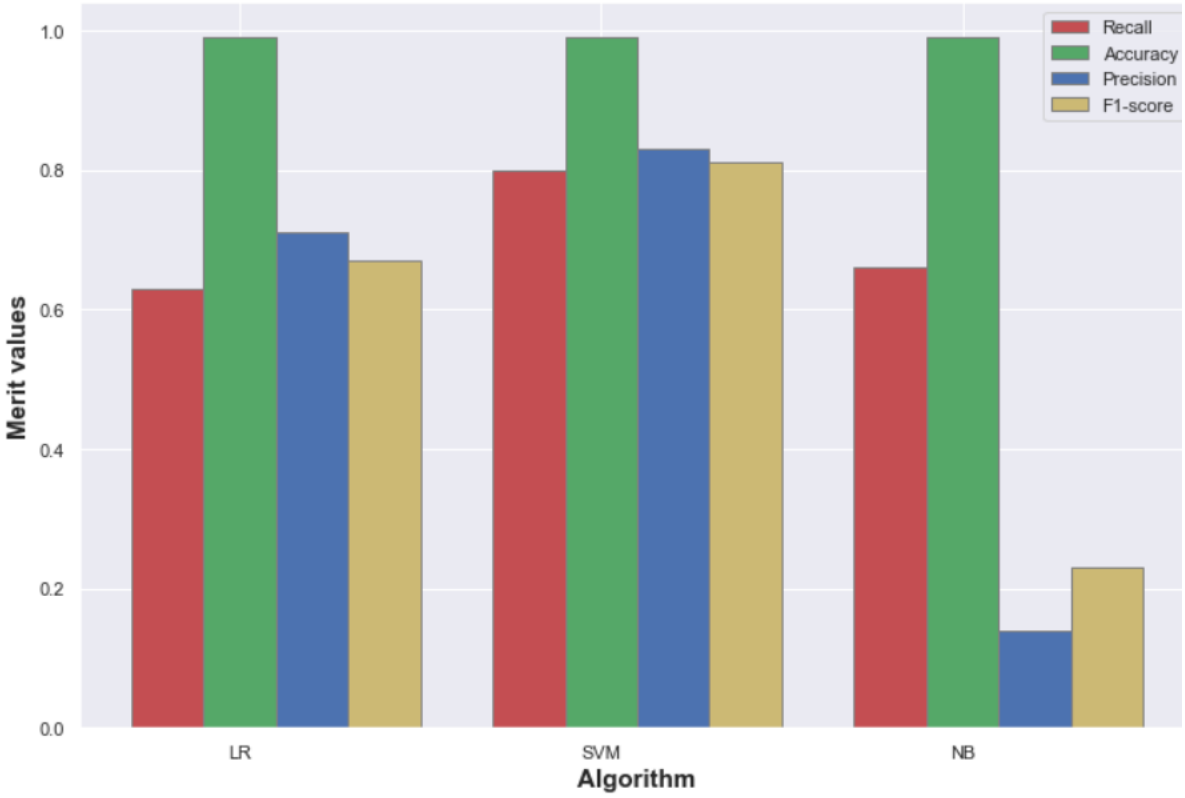


Heatmap also suggests that the data is highly imbalanced.

Confusion matrix for Support Vector Machines:



Comparison of metric values for each algorithm applied:



At the end we can see that SVM was the best among the three algorithms to detect the fraud.

References:

[1] W. Richert, L. P. Coelho, "Building Machine Learning 978-1-78216-140-0

[2]] S. Marsland, Machine learning: an algorithmic perspective. CRC press, 2015

Author Biography

VELISETTI GEETHA PAVAN SAHASRANTH is a final year student of B.Tech. Degree in Computer and Communication Engineering from MIT Manipal, Manipal Academy of Higher Education (Institution of Eminence), India. His research interests include applying artificial intelligence technology to practical problems, data science, software testing and debugging.