REAL TIME APPROACH TO LOAN CREDIT

USING SCF DATASET

Sravanii Gottimukkala1, Nidhi2 and Ankit A. Jain3

1 Rajiv Gandhi University of Knowledge Technologies, IIIT RK Valley

2

3

**Abstract**

The main goal of this study is to establish a new benchmark using real consumer data and to provide machine learning approaches to decide whether the consumer is eligible to take the loan or not. We performed machine learning approaches on the Survey of Consumer Finances (SCF) data. SCF data is non-synthetic and consists of a large number of real variables so we applied feature selection techniques like Chi-square test, Random Forest and Boruta algorithms for effective modeling. We then trained the new obtained limited features with machine learning algorithms like Decision Tree, Random Forest, XG Boost and Logistic Regression and calculate the accuracy, precision, recall, F1-Score and area under the curve (AUC), respectively.

Keywords: Chi-square test, Random forest, Boruta, Decision Tree, XG Boost, Logistic Regression.

**1. Introduction**

For lending institutions, credit scoring systems aim to provide probability of default (PD) for their clients and to satisfy a minimum-loss principle for their sustainability. Therefore, a credit scoring system supports decision making for credit applications, manages credit risks and inﬂuences the amount of non-performing loans that are likely to lead to bankruptcy, ﬁnancial crisis and environment sustainability [1].

For this we performed machine learning approaches on the Survey of Consumer Finances (SCF) data. SCF data [2] is non-synthetic and consists of a large number of real variables so we applied feature selection techniques [3].

In a high dimensional dataset, there remain some entirely irrelevant, insignificant and unimportant features. It has been seen that the contribution of these types of features is often less towards predictive modeling as compared to the critical features. They may have zero contribution as well. These features cause a number of problems which in turn prevents the process of efficient predictive modeling –

• Unnecessary resource allocation for these features.

• These features act as a noise for which the machine learning model can perform terribly poorly.

• The machine model takes more time to get trained.

So, the solution for this is **Feature Selection [4]**.

Feature Selection is the process of selecting out the most significant features from a given dataset. In many of the cases, Feature Selection can enhance the performance of a machine learning model as well.

Feature selection is also known as Variable selection or Attribute selection.

Essentially, it is the process of selecting the most important/relevant features of a dataset.

**Importance of Feature Selection:**

The importance of feature selection can best be recognized when you are dealing with a dataset that contains a vast number of features. This type of dataset is often referred to as a high dimensional dataset. Now, with this high dimensionality, comes a lot of problems such as - this high dimensionality will significantly increase the training time of your machine learning model, it can make your model very complicated which in turn may lead to overfitting.

Often in a high dimensional feature set, there remain several features which are redundant means these features are nothing but extensions of the other essential features. These redundant features do not effectively contribute to the model training as well. So, clearly, there is a need to extract the most important and the most relevant features for a dataset in order to get the most effective predictive modeling performance.

**Summary of Importance of Feature Selection:**

• It enables the machine learning algorithm to train faster.

• It reduces the complexity of a model and makes it easier to interpret.

• It improves the accuracy of a model if the right subset is chosen.

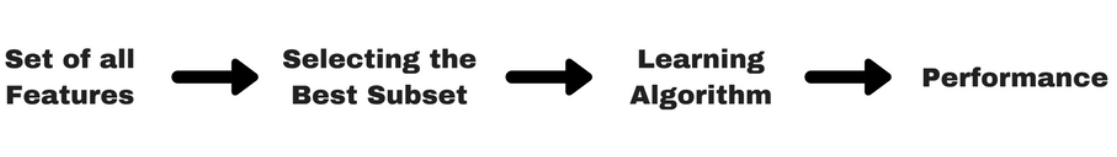
• It reduces Overfitting.

The rest of the paper is organized as follows, section 2 summarizes techniques of Feature Selection. The dataset and description for the obtained variables after feature selection are discussed in section 3. Section 4 details the analysis of results obtained. Finally Section 5 concludes the paper with important observations.

**2. Feature Selection Methods**

There are a few techniques of feature selection:

* Filter Methods,
* Wrapper Methods,
* Tree-based feature importance,
* Sophisticated algorithms such as Boruta.

**(1) Filter Method:**

Filter method [5] relies on the general uniqueness of the data to be evaluated and pick feature subset, not including any mining algorithm. Filter method uses the exact assessment criterion which includes distance, information, dependency, and consistency. The filter method uses the principal criteria of ranking technique and uses the rank ordering method for variable selection. The reason for using the ranking method is simplicity, produce excellent and relevant features. The ranking method will filter out irrelevant features before classification process starts.

Filter methods are generally used as a data preprocessing step. The selection of features is independent of any machine learning algorithm. Features give rank on the basis of statistical scores which tend to determine the features' correlation with the outcome variable. Correlation is a heavily contextual term, and it varies from work to work.

**Examples of Filter Method:**

**(a) Pearson Correlation:**

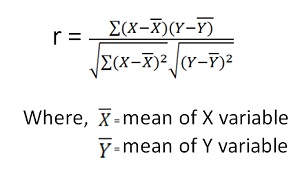
Here we will first plot the Pearson correlation [6] heatmap and see the correlation of independent variables with the output variable. We will only select features which has correlation of above 0.5 (taking absolute value) with the output variable.

The correlation coefficient has values between -1 to 1.

— A value closer to 0 implies weaker correlation (exact 0 implying no correlation).

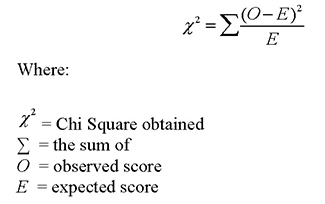
— A value closer to 1 implies stronger positive correlation.

— A value closer to -1 implies stronger negative correlation.



**(b) Chi – Square Test:**

A chi-square test [6] is used in statistics to test the independence of two events. Given the data of two variables, we can get observed count O and expected count E. Chi-Square measures how expected count E and observed count O deviates each other.

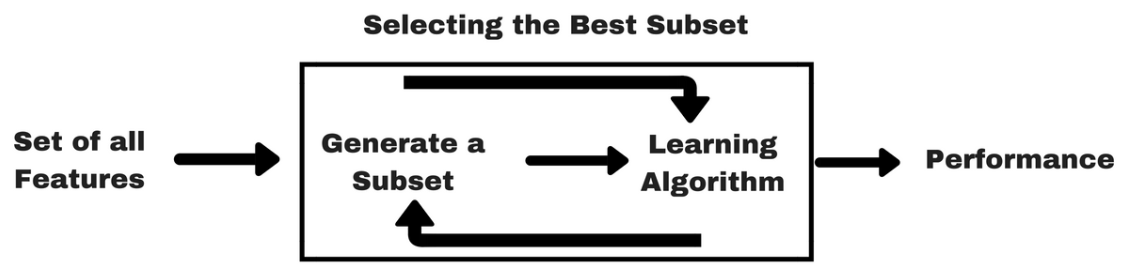


In feature selection, we aim to select the features which are highly dependent on the response (dependent variable).

When two features are independent, the observed count is close to the expected count, thus we will have smaller Chi-Square value. So high Chi-Square value indicates that the hypothesis of independence is incorrect. In simple words, higher the Chi-Square value the feature is more dependent on the response and it can be selected for model training.

**(2) Wrapper Method:**

A wrapper method [7] needs one machine learning algorithm and uses its performance as evaluation criteria. This means, you feed the features to the selected Machine Learning algorithm and based on the model performance you add/remove the features. This is an iterative and computationally expensive process but it is more accurate than the filter method.



**Example of Wrapper Method:**

**Recursive Feature Elimination (RFE):**

Recursive feature elimination [8] performs a greedy search to find the best performing feature subset. It iteratively creates models and determines the best or the worst performing feature at each iteration. It constructs the subsequent models with the left features until all the features are explored. It then ranks the features based on the order of their elimination. In the worst case, if a dataset contains N number of features RFE will do a greedy search for 2^N combinations of features.

**(3) Tree-based Feature Importance:**

After training of tree ensemble methods [9] such as Random Forests or Extra Tree, we can access relative importance of each feature. We can say that it is a by-product of these tree-based estimators. The values of future importance can be then used directly to perform feature selection.

**Random Forest:**

Random Forests [10] are often used for feature selection in a data science workflow. The reason is because the tree-based strategies used by random forests naturally ranks by how well they improve the purity of the node. This mean decrease in impurity over all trees (called gini impurity). Nodes with the greatest decrease in impurity happen at the start of the trees, while notes with the least decrease in impurity occur at the end of trees. Thus, by pruning trees below a particular node, we can create a subset of the most important features.

**Steps:**

1. Prepare the dataset.

2. Train a random forest classifier.

3. Identify the most important features.

4. Create a new ‘limited featured’ dataset containing only those features.

5. Train a second classifier on this new dataset.

6. Compare the accuracy of the ‘full featured’ classifier to the accuracy of the ‘limited featured’ classifier.

**(4) Boruta:**

In contrary to the previous algorithms, Boruta [11] tries to find all relevant features useful for prediction, instead of defining a subset of features with minimal error. By default, Boruta uses Random Forest.

The following reasons to use Boruta package for feature selection:

1. It works well for both classification and regression problem.

2. It takes into account multi-variable relationships.

3. It is an improvement on random forest variable importance measure which is a very popular method for variable selection.

4. It follows an all-relevant variable selection method in which it considers all features which are relevant to the outcome variable. Whereas, most of the other variable selection algorithms follow a minimal optimal method where they rely on a small subset of features which yields a minimal error on a chosen classifier.

5. It can handle interactions between variables.

6. It can deal with fluctuating nature of random a random forest importance measure.

**Basic idea of Boruta Algorithm:**

Perform shuffling of predictors (input) values and join them with the original predictors and then build random forest on the merged dataset. Then make comparison of original variables with the randomized variables to measure variable importance. Only variables having higher importance than that of the randomized variables are considered important.

**Steps of Boruta Algorithm:**

1. Create duplicate copies of all independent variables. When the number of independent variables in the original data is less than 5, create at least 5 copies using existing variables.

2. Shuffle the values of added duplicate copies to remove their correlations with the target variable. It is called shadow features or permuted copies.

3. Combine the original ones with shuffled copies.

4. Run a random forest classifier on the combined dataset and performs a variable importance measure (the default is Mean Decrease Accuracy) to evaluate the importance of each variable where higher means more important.

5. Then Z score is computed. It means mean of accuracy loss divided by standard deviation of accuracy loss.

6. Find the maximum Z score among shadow attributes (MZSA).

7. Tag the variables as 'unimportant' when they have importance significantly lower than MZSA. Then we permanently remove them from the process.

8. Tag the variables as 'important' when they have importance significantly higher than MZSA.

9. Repeat the above steps for predefined number of iterations (random forest runs), or until all attributes are either tagged 'unimportant' or 'important', whichever comes first.

**3. SCF Dataset:**

We use SCF 2016 (Survey of Consumer Finances) Dataset. The dataset is retrieved from The Federal Reserve’s normally triennial cross-sectional survey of U.S. families. SCF consists of information about families’ balance sheets, pensions, income, demographic characteristics and the borrower’s attitude. SCF dataset had established an excellent foundation for the household payment problem. Therefore, this dataset is more suitable for the investigation of techniques and methodologies of credit scoring. SCF (2016) data contains 348 variables. The SCF survey started to provide information obtained from borrowers about their debt repayment behavior. Prior to the SCF, most information about delinquent debt repayment came from lenders. Therefore, we choose delinquent debt repayment variable (LATE) as dependent variable. If a household had no late debt payments, the LATE variable is “no” and 0. Otherwise, LATE variable is “yes” and 1.

We got good results with Random Forest feature selection method. It gives 13 important features (variables). The following will give the description for those obtained variables.

1) LATE60 – Delinquent debt repayment variable. It describes whether household had debt payment more than 60 days past due in last year.

2) LIQ - Total value of all types of transactions accounts, 2016 dollars.

3) FEARDENIAL - Household feared being denied credit in the past 5 years.

4) NETWORTH - Total net worth of household, 2016 dollars.

5) FIN - Total value of financial assets held by household, 2016 dollars.

6) ASSET - Total value of assets held by household, 2016 dollars.

7) TURNFEAR - Household has been turned down for credit or feared being denied credit in the past 5 years.

8) LEVRATIO - Ratio of total debt to total assets.

9) Y1 - Case ID with implicate number.

10) YY1 - Case ID.

11) CHECKING - Total value of checking accounts held by household, 2016 dollars.

12) PIRTOTAL - Ratio of monthly debt payments to monthly income.

13) DEBT2INC - Ratio of total debt to total income.

**4. Results:**

We estimate models built over various machine-learning algorithms and feature-selection algorithms. But with Random Forest feature selection algorithm good results are found.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| MODEL | Train\_Accuracy | Test\_Accuracy | Precision | Recall | F1\_Score | ROC\_AUC |
| Decision Tree | 1 | 0.981783 | 0.933148 | 0.947666 | 0.940351 | 0.967771 |
| Random Forest | 1 | 0.992285 | 0.998514 | 0.950495 | 0.973913 | 0.975121 |
| XG Boost | 0.952197 | 0.940206 | 0.993088 | 0.609618 | 0.755478 | 0.80443 |
| Logistic Regression | 0.91313 | 0.914702 | 1 | 0.437058 | 0.608268 | 0.718529 |

**5. Conclusion:**

In this paper we discuss the importance of feature selection techniques for better performance. We built machine learning algorithms to estimate borrowers’ probability of default. Among all the algorithms we got good results with the Random Forest algorithm.

**6. Reference:**

[1] Munkhdalai, Lkhagvadorj, et al. "An empirical comparison of machine-learning methods on bank client credit assessments." *Sustainability* 11.3 (2019): 699.

[2] Bucks, B.K.; Kennickell, A.B.; Moore, K.B. Recent changes in US family ﬁnances: Evidence from the 2001 and 2004 Survey of Consumer Finances. Fed. Res. Bull. 2006, A1, 92.

[3] Guyon, Isabelle, and André Elisseeff. "An introduction to variable and feature selection." Journal of machine learning research 3.Mar (2003): 1157-1182.

[4] Sayak Paul, *Beginner's Guide to Feature Selection in Python,* Datacamp, January 2020.

[5] Renu Khandelwal, *Feature selection in Python using the Filter method,* October 2019.

[6] Rahul Agarwal, *The 5 Feature Selection Algorithms every Data Scientist should know*, July 2019.

[7] Jason Browniee, *An Introduction to Feature Selection - Machine Learning Mastery,* October 2014.

[8] Dario*, Feature Selection in Python — Recursive Feature Elimination,* September 2019.

[9] Dietterich, Thomas G. "Ensemble methods in machine learning." International workshop on multiple classifier systems. Springer, Berlin, Heidelberg, 2000.

[10] Chris Albon, *Feature Selection Using Random Forest,* December 2017.

[11] Deepanshu Bhalla, *Select Important Variables using Boruta Algorithm,* June 2017.