Analysis of Clustering, Classification, and Pattern Mining Techniques on Consumer Credit Risk Data

Botond A. Nás

ELTE Faculty of Informatics,
Pázmány Péter stny. 1/C., 1117 Budapest, HU
khupib@inf.elte.hu
https://www.inf.elte.hu/en/

Abstract. In a world saturated with information, smart systems are needed to detect patterns and make predictions using large quantities of data. Such systems can benefit businesses, such as banks, in making operational decisions. In this report, we compare the performance of k-means and k-prototype clustering in grouping consumers as 'good' or 'bad' using a data set related to consumer credit risk. We use the traditional Rand index and Jaccard index as performance measures to compare the two clustering methods. We also train five classification models, each using a different classification method, and compare their ability to accurately predict whether a consumer is 'good' or 'bad'. We observe the Area Under the Receiver Operating Characteristics Curve and the confusion matrices, as well as the F1 score to evaluate each method. We observe patterns in the data using the Apriori Frequent Pattern Mining algorithm and discuss how association rules can be used to improve classification. Our experiments indicate that the k-prototypes method slightly outperforms the k-means method for clustering and the Logistic Regression classification method yields better predictions than the other tested methods for this data set.

Keywords: Clustering · Classification · Frequent Pattern Mining

1 Introduction

Classification and clustering are fundamental operations in data mining and fall into the categories of supervised and unsupervised learning respectively. Unsupervised learning algorithms learn and detect patterns from unlabelled data. In supervised learning, on the other hand, the goal is to, given a training set, learn a function h so that $h(x)$ is a "good" predictor for the corresponding value of y [11]. When the target variable is continuous it is called a *regression* learning problem and when there are only a small number of discrete values it can take, it is called a *classification* problem. In this report, we compare the performance of various classification and clustering methods on a data set related to consumer credit risk and use the Apriori Frequent Pattern Mining (FPM) algorithm to uncover

patterns and rules in the data set. The data contains numerical and categorical data and the target values are binary, i.e. consumer is either 'good' or 'bad'. These characteristics of the data heavily influence the selected clustering and classification methods. We investigate the performance of k-means clustering and its variant, k-prototypes, in grouping consumers as 'good' or 'bad' based on the data. We also develop five classification models, train them with a training set of the data, and compare their performance in predicting the consumer classification (detecting 'bad' consumers) on a test set of the data. The classification methods compared in this report are: Logistic Regression, Gaussian Naive Bayes, k-Nearest Neighbors, Random Forest, and Support Vector Machines (SVM). The data is further inspected using the Apriori algorithm to detect association rules.

2 Methods

The data processing and analysis was done using Python with the help of the `scikit-learn`, `NumPy`, `Pandas`, `apyori` [2], `matplotlib`, and `kmodes` [3] libraries.

2.1 Data Preprocessing

Before it can be used, the data must be understood and prepared. The data consists of a combination of nominal, quantitative, and boolean attribute types. In order to fit and evaluate a classification model to the data, it had to be scaled and encoded. For clustering, the preprocessing varied depending on the method employed. K-means clustering required scaling and encoding while k-prototypes clustering only required scaling for numeric attributes. The Apriori algorithm required neither.

After the data set was checked for missing values, the quantitative features were scaled using Standardization (Z-score normalization). Z-score normalization standardizes features by subtracting the mean and scaling to unit variance:

$$z = \frac{x - \mu}{\sigma} \tag{1}$$

Normalization is necessary to make every data point have the same scale so each feature is equally important. The nominal features were encoded using One-Hot encoding to convert the categorical data to numeric types. One-Hot encoding creates binary columns, indicating the presence of each possible value from the original data. The label column contains either a 1 or a 2, indicating whether the customer is 'good' or 'bad', respectively. Label encoding was used to convert these class labels to binary 0 or 1 to fit the binary classification schema of the task. Before the models were created, the data was split into training and test sets with a 4:1 training to test ratio. Due to the lack of balance in the data set, stratified data splitting was used to split the data in a way that preserved the same proportions of examples in each class as observed in the original data set.

2.2 Unsupervised Learning for Clustering

Unsupervised algorithms make inferences from data sets using only input vectors without referring to known, or labelled, outcomes. Such methods are employed for clustering. The aim of clustering is to group objects into clusters, such that objects in the same cluster are more similar to each other than to the objects belonging to other clusters [8]. Similarity, $s(x, y) \in [0, 1]$ for $x, y \in D$, is highly important in clustering. Due to the mixed attribute-type nature of the data set, two clustering methods were utilized and compared: k-means clustering with One-Hot encoded data and k-Prototype clustering.

k-Means Clustering The k-Means algorithm identifies k number of centroids, and then allocates every data point to the nearest cluster, while keeping the centroids as small as possible. The 'means' in k-Means refers to averaging of the data; that is, finding the centroid. Given $D \subseteq D_1 \times D_2 \times \cdots \times D_m$, a distance measure d, and the number of clusters k, k-Means clustering finds the cluster centers $\mathrm{c}_1, \mathrm{c}_2, ..., \mathrm{c}_k$ and a mapping $p : D \to \{1, 2, ..., k\}$ such that

$$\sum_{i=1}^{n} d(\mathrm{x}_i, \mathrm{c}_{p(\mathrm{x}_i)}) \tag{2}$$

is minimized [8]. The distance metric used was the Euclidean distance, defined as:

$$d(\mathrm{p}, \mathrm{q}) = \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2} \tag{3}$$

, where p, q are two points in Euclidean n-space and q_i, p_i are Euclidean vectors starting from the origin. The k-Means algorithm isn't directly applicable to categorical data because the sample space for categorical data is discrete and doesn't have a natural origin. Therefore, the categorical columns of the data were One-Hot encoded, converting them to numeric types.

Algorithm 1: k-Means Clustering

Result: p and $\mathrm{c}_1, \mathrm{c}_2, ..., \mathrm{c}_k$
Choose the number of clusters k;
Initialize centroids $\mathrm{c}_1, \mathrm{c}_2, ..., \mathrm{c}_k$ randomly (random initialization);
repeat
  for *each data point* x_i **do**
    find the nearest centroid, c_p, by finding p such that (2) is minimal;
    assign the point to that cluster;
  end
  for *each* c_j **do**
    update centroid: $\mathrm{c}_j = \frac{\sum_{\mathrm{x}, p(\mathrm{x})=j} \mathrm{x}}{\sum_{\mathrm{x}, p(\mathrm{x})=j} 1}$;
  end
until *convergence*;

k-Prototypes Clustering In 1998, Huang [10] introduced two extensions to the k-means algorithm, making it suitable for categorical values and mixed, numeric and categorical, data sets. The first, k-modes, extends the k-means algorithm to cluster categorical data by using a simple matching dissimilarity measure for categorical objects, modes instead of means for clusters and a frequency-based method to update modes in the k-means fashion clustering process to minimize the clustering cost function [10]. The dissimilarity measure, known as simple matching, between two categorical values X and Y is defined by Huang [10] as

$$d(X,Y) = \sum_{j=1}^{m} \delta(x_j, y_j), \text{ where } \delta(x_j, y_j) = \begin{cases} 0 & \text{if } x_j = y_j \\ 1 & \text{if } x_j \neq y_j \end{cases} \tag{4}$$

The k-prototypes algorithm integrates the k-means and k-modes processes to cluster data with mixed numeric and categorical values. In the k-prototypes algorithm a dissimilarity measure is defined that takes into account both numeric and categorical attributes [10]. The dissimilarity between two mixed-type objects X and Y is measured by

$$d(X,Y) = \sum_{j=1}^{p} (x_j - y_j)^2 + \gamma \sum_{j=p+1}^{m} \delta(x_j, y_j) \tag{5}$$

where the first term is the squared Euclidean distance measure on the numeric attributes and the second term is the simple matching dissimilarity measure on the categorical attributes [10]. The weight γ is used to avoid favoring either type of attribute.

2.3 Supervised Learning for Classification and Prediction

One of the main roles served by a classification model is to be used as a predictive model to classify previously unlabeled instances [13]. The no Free Lunch Theorem [14] states, however, that there is no one best algorithm that works the best in all cases. This section details the five different classification methods investigated in this report for predictive modelling.

Logistic Regression In binary classification, y can only take on two values, 0 and 1. A logistic function is a sigmoid function that takes any real input and outputs a value between zero and one and is defined as

$$g(z) = \frac{1}{1 - e^{-z}} \tag{6}$$

The function, $g(z)$ approaches one as z approaches infinity and zero as z approaches negative infinity. Thus, the hypothesis is defined as

$$h_\theta(x) = \frac{1}{1 - e^{-\theta^T x}} \tag{7}$$

The output from the hypothesis is the estimated probability. Its value corresponds to the confidence that the predicted value can be the actual value. In other words, it represents the probability that $y = 1$ given x which is parameterized by θ:

$$P(y = 1|x; \theta) = h_\theta(x)$$
$$P(y = 0|x; \theta) = 1 - h_\theta(x)$$

Data is fit into the linear regression model via maximum likelihood estimation, which is then acted upon by a logistic function predicting the target categorical dependent variable [11].

Gaussian Naive Bayes Naive Bayes classifiers are a collection of classification algorithms based on Bayes' Theorem:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{8}$$

Bayes' Theorem finds the probability of an event occurring given the probability of another event that has already occurred. Naive Bayes is naive because it assumes that the value of a particular feature is independent of the value of any other feature, given the class variable. Using this assumption, the following classification rule can be used:

$$\hat{y} = \arg\max_y P(y) \prod_{i=1}^{n} P(x_i|y) \tag{9}$$

, given class variable y and dependent feature vector $x_1, ..., x_n$ [1]. Gaussian naive Bayes assumes that the continuous values associated with each class are distributed according to a normal (or Gaussian) distribution. Thus, the likelihood of the features is assumed to be Gaussian:

$$P(x_i|y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}\right) \tag{10}$$

The parameters σ_y and μ_y are estimated using maximum likelihood [1].

k-Nearest Neighbors K-Nearest Neighbors is the simplest of the investigated methods. It relies solely on the class of the nearest neighbors of a data point to predict its class. The integer k, a user-defined constant, constitutes how many neighbors are considered. High k values suppress noise but make the classification boundaries less distinct. Common distance metrics used include the Euclidean distance and Manhattan distance, generalized by the Minkowski distance. The power parameter p determines the distance metric. The hyperparameters p=1 and k=15 were determined using `scikit-learn`'s GridSearch tool.

Random Forest Decision Tree classification creates a training model that can be used to predict the class or value of the target variable by learning simple decision rules inferred from the training data. However, trees constructed with fixed training data are prone to overly adapt to the training data [7]. Random Forests are ensemble classification methods that are a combination of tree predictors. Each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest [5].

Support Vector Machines (SVM) Support Vector Machine (SVM) is another popular classification algorithm, although it can be used for regression as well [12]. The objective of the support vector machine algorithm is to find a hyperplane in an N-dimensional space (N number of features) that distinctly classifies the data points [6]. To separate two classes of data points, there are many possible hyperplanes that could be chosen. SVM finds a plane that has the maximum margin, i.e the maximum distance between data points of both classes. Hyperplanes are decision boundaries that classify the data points and their dimension depends upon the number of features. Support vectors are data points that are closer to the hyperplane and influence the position and orientation of the hyperplane. These support vectors are used to maximize the margin of the classifier [6].

Given a training set of instance-label pairs $(x_i, y_i), i = 1, ..., l$ where $x_i \in R^n$ and $y \in \{-1, 1\}^l$, the SVM require the solution of the following optimization problem [9]:

$$\begin{aligned} \min_{\mathrm{w},b,\xi} \ & \frac{1}{2}\mathrm{w}^T\mathrm{w} + C\sum_{i=1}^{l}\xi_i \\ \text{subject to } & y_i(\mathrm{w}^T\phi(\mathrm{x}_i) + b) \geq 1 - \xi_i, \\ & \xi_i \geq 0 \end{aligned}$$

The training vectors x_i are mapped to a higher dimensional space by the function ϕ. SVM finds a linear separating hyperplane with the maximal margin in this higher dimensional space. $C > 0$ is the penalty parameter of the error term. Furthermore, $K(\mathrm{x}_i, \mathrm{x}_j) \equiv \phi(\mathrm{x}_i^T)\phi(\mathrm{x}_j)$ is called the kernel function [9]. The polynomial kernel function was used, defined as:

$$\begin{aligned} & K(\mathrm{x}_i, \mathrm{x}_j) = (\gamma \mathrm{x}_i^T \mathrm{x}_j + r)^d, \gamma > 0 \\ & \text{where } \gamma, r, \text{ and } d \text{ are kernel parameters.} \end{aligned}$$

The hyperparameters $C = 0.1$, $d = 1$, $\gamma = 1$ were determined using `scikit-learn`'s GridSearch tool.

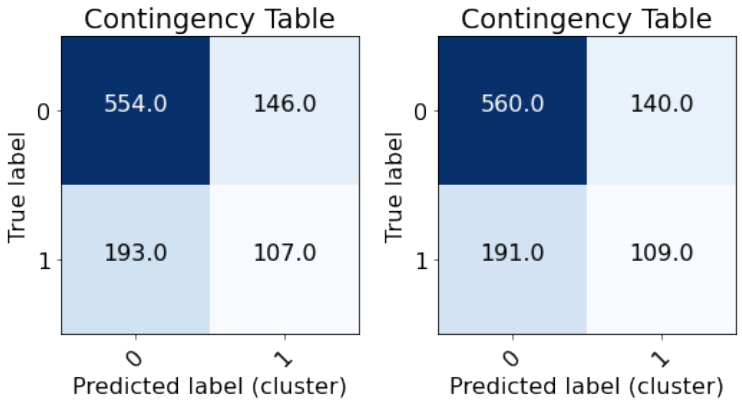
2.4 Frequent Pattern Mining

The Apriori algorithm was used to find underlying relations between different items using the `apyori` [2] Python package's implementation of the algorithm.

The Apriori algorithm uses frequent item sets to generate association rules and, using these association rules, it determines how strongly or how weakly two objects are connected. There are three main components of Apriori: support, confidence, and lift. Support is the popularity of an item (value) and is calculated by finding the number of transactions (rows) containing a particular item divided by total number of transactions (rows). Confidence refers to the likelihood that a value X occurs (is true) if another value Y is true. It can be calculated by finding the number of rows where X and Y both occur, divided by total number of rows where X occurs. Lift is calculated by dividing confidence by support and represents the increase in the ratio of the occurrence of value Y when X occurs. A lift of 1 means there is no association between values X and Y and a lift greater than 1 means values X and Y are more likely to occur together. A lift of less than 1 refers to the case where two values are unlikely to occur together. The algorithm was executed on the data set with a minimum support of 0.05, a minimum confidence of 0.6, and a minimum lift of 2.

3 Results and Discussion

To evaluate the clustering methods, a contingency table (Figure 1) was used to determine the Rand index and Jaccard index of the models. The value 0 represents 'good customer' and 1 represents 'bad customer'.



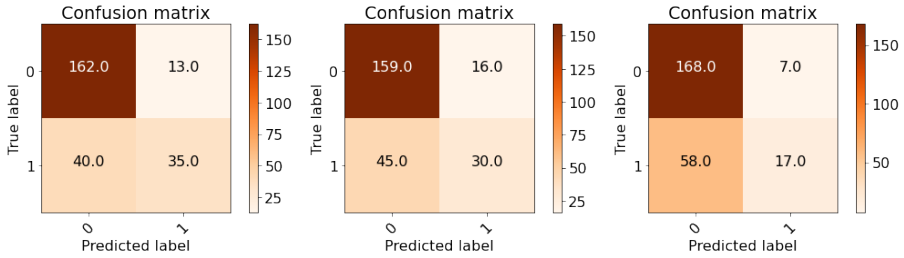
(a) k-means (b) k-prototypes

Fig. 1: Contingency tables

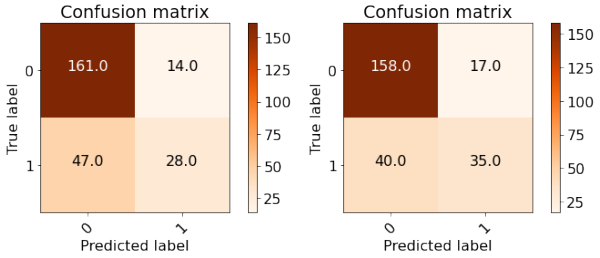
The Rand index values for the k-means and k-prototypes models were 0.661 and 0.669 respectively and the Jaccard index values were 0.239 and 0.247.

The Area Under the ROC (Receiver Operating Characteristics) Curve (AUC) was used as a performance measure for the classification models, as well as the

confusion matrix (Figure 2) for visualization and its corresponding classification measures (Table 1). Using the information provided by the confusion matrices, the weighted average recall and precision were calculated. Recall provides information about a classifier's performance with respect to false negatives, while precision provides information about a classifier's performance with respect to false positives. The F1 score can be derived from the precision and recall, giving an all encompassing metric. The Logistic Regression model yielded the highest weighted average F1 score of 0.77, followed by the SVM method with 0.76. These two models outperformed the other models in true negative prediction. The KNN method had the lowest weighted average F1 score of .69 and the Gaussian Naive Bayes and Random Forest methods yielded 0.74 and 0.73 respectively.



(a) Logistic Regression (b) Gaussian Naive Bayes (c) KNN



(d) Random Forest (e) SVM

Fig. 2: Confusion matrices

ROC is a probability curve and AUC represents the degree of separability, or how much the model is capable of distinguishing between classes. A model with higher AUC is better at predicting the true value. The AUC values yielded a similar ranking as the weighted average F1 values. It is important to note, however, that ROC averages over all possible thresholds, making it inferior to the F1 score when there is data imbalance.

The results of the Apriori FPM algorithm can be used to improve the classification and predication models by finding the attributes most associated with consumer credit risk. Our configuration highlighted a clear association between the job situation and living accommodation of a consumer and their sex, foreign

Classification Method	Precision	Recall/Sensitivity	AUC
Logistic Regression	0.78	0.79	0.81
Gaussian Naive Bayes	0.74	0.76	0.79
KNN	0.73	0.74	0.76
Random Forest	0.74	0.76	0.80
SVM	0.76	0.77	0.79

Table 1: Classification performance measures

worker status, and property ownership (Table 2). Consumers living for free that were skilled workers or officials were often single male foreign workers with no known property ownership and no co-applicant or guarantor on their credit. Also, 'bad' consumers that lived for free were often foreign single male workers who didn't own any property and had no co-applicant or guarantor on their credit. With more time and analysis of the rules, additional valuable patterns can be detected in the data set. FPM can be applied to classification to improve the classification model using the patterns and rules discovered by the FPM algorithm. By combining the patterns and class variables in the rules, discriminative patterns can be found for the purpose of classification [4].

Item Set 1	Item Set 2	Support	Confidence	Lift
(A153, A173)	(A201, A101, A124, A93)	0.050	0.793651	7.631258
(2, A153)	(A101, A201, A124, 4, A93)	0.57	0.662791	7.618284
(2, A153)	(A101, A124, 4, A93)	0.57	0.662791	7.531712
(2, A153)	(A101, A124, A61)	0.53	0.616279	7.515598
(2, A153)	(A201, A101, A124, A61)	0.53	0.616279	7.515598

A153	consumer lives for free
A173	consumer is a skilled employee/official
2	consumer is a 'bad' consumer
A201	consumer is a foreign worker
A101	consumer has no other debtors or guarantors for the credit
A124	consumer doesn't own property or property ownership is unknown
A93	consumer is a single male
4	installment rate is 4 percentage of consumer's disposable income
A61	consumer has less then 100 EUR in savings/bonds

Table 2: Top 5 association rules (ordered by lift)

4 Conclusion

The two clustering methods clustered the data similarly, with k-prototypes performing a fraction better than k-means (with encoded data). The minimal superiority of k-prototypes shows that the use of different distance metrics for

categorical and numeric data determines similarity with higher accuracy. The low Jaccard indices of the two methods show that when only considering true positives, the prediction was quite dissimilar to the true labels. In order to find the optimal clustering algorithm for this data set, more techniques, such as Hierarchical clustering, must be investigated. For classification and prediction, Logistic Regression proved to be the most effective and KNN the least effective for this data set. Despite it's low weighted average F1 score, the KNN method yielded the least false positives and the most true negatives. This is most likely due to the unbalanced nature of the data; there are more 'good' customers in the data set than 'bad'. In order to conduct a more fair and definitive comparison, more precise optimal hyperparameters need to be determined for each method. The Apriori FPM method provided valuable insight into the patterns present in the data. Applying these findings to classification could improve prediction accuracy and would help in developing the optimal classification model for this data set.

References

1. 1.9. naive bayes. https://scikit-learn.org/stable/modules/naive_bayes.html, accessed: 2021–05-21
2. apyori, https://pypi.org/project/apyori/
3. kmodes, https://pypi.org/project/kmodes/
4. Aggarwal, C.C., Han, J.: Frequent Pattern Mining. Springer International Publishing, 1 edn. (2014)
5. Breiman, L.: Random forests. Machine Learning **45**(1), 5–32 (2001). https://doi.org/10.1023/A:1010933404324, http://dx.doi.org/10.1023/A%3A1010933404324
6. Gandhi, R.: Support vector machine - introduction to machine learning algorithms (Jul 2018), https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47
7. Ho, T.K.: Random decision forests. Proceedings of 3rd International Conference on Document Analysis and Recognition **1**, 278–282 (Aug 1995). https://doi.org/10.1109/icdar.1995.598994
8. Horvath, T.: Lecture 2: Clustering (February 2021)
9. Hsu, C.w., Chang, C.c., Lin, C.J.: A practical guide to support vector classification chih-wei hsu, chih-chung chang, and chih-jen lin (11 2003)
10. Huang, Z.: Extensions to thek-means algorithm for clusteringlarge data sets with categorical values. Data Mining and Knowledge Discovery **2**, 283–304 (1998), http://www.cs.ust.hk/~qyang/Teaching/537/Papers/huang98extensions.pdf
11. Ng, A.: Cs229 lecture notes - supervised learning (May 2021), https://akademik.bahcesehir.edu.tr/~tevfik/courses/cmp5101/cs229-notes1.pdf
12. Ray, S.: Svm: Support vector machine algorithm in machine learning (Oct 2015), https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/
13. Tan, P.N., Steinbach, M., Karpatne, A., Kumar, V.: Introduction to Data Mining. Pearson (2020)
14. Wolpert, D., Macready, W.: No free lunch theorems for optimization. IEEE Transactions on Evolutionary Computation **1**(1), 67–82 (1997). https://doi.org/10.1109/4235.585893