ISSN No: 2456-2156

Performance Evaluation of Svm in a Real Dataset to Predict Customer Purchases

Ledion Lico
Gener2 SH.P.K
ABA Tower
Tirana, Albania

Indrit Enesi
Electronic and Telecommunication Department
Polytechnic University of Tirana
Tirana, Albania

Abstract:- Predicting Customer behavior is key to marketing strategies. Customer Relationship Management technology plays a very important role in business performance. Predicting customer behavior enables the business to better address their customers and enhance service level and overall profit. A model based on Support Vector Machines is proposed used to classify clients and predict their purchases in a real retail department store. Different Kernels functions are used and their performance is evaluated. The data scaling is implemented with SVM model and its performance is evaluated. The study is conducted in a real retail department store in Albania for the 2020 year. Implementations in python shows that the proposed model performs better in time and accuracy.

Keywords:- *Classification; Cross-Validation; Machine Learning; Neural Networks; SVM, Standard Scale Function.*

I. INTRODUCTION

According to Forbes, the disruptive impact of artificial intelligence in retail is seen across the value chain and is emerging as a powerful tool for retail brands to gain a strategic advantage over their competition. Marketing research firm Tractica has predicted that global AI revenues will grow from $643.7 million in 2016 to an excess of $36.8 billion in 2025 [1]. Customer behavior prediction is a key concept that helps companies better manage their stocks and adjust marketing strategies. Machine Learning Algorithms can be applied to predict customer purchases in a retail transactions database. The challenge remains building the right model using the right algorithm and the right predictors. Various models have been proposed to perform this task. SVM has been used across different industries to predict customer churn with good results [2][6]. It has also been used to predict customer behavior in online shopping using time spent in the website and the purchasing patterns as predictors [4].The authors did not supply accuracy metrics and used only the linear kernel.

The model that we propose in this paper will be based on Support Vector Machines and will have 6 different predictors including the attributes of the client and his purchasing history (if any) and will be applied and tuned to a real retail dataset with the aim to predict the customer label (class) at the end of the year based on quarterly purchases data. The model will be applied to both the unscaled and scaled data and the performance will be evaluated. Different Kernel Functions will be implemented and compared based on the accuracy produced. The dataset consists of 3797 records of client's purchases of the first quarter of 2020 in a real department store.

Paper is organized as follows: in the section 2 SVM classification method with different kernels is analyzed. Section 3 discusses about Standard Scaler Function. Implementations and results are shown in section 4. Paper closes with conclusions and future work.

II. SUPPORT VECTOR MACHINES

Support Vector Machines (SVMs) is an optimal classification method by maximizing the margin between classes. For n points in a d-dimensional space $\boldsymbol{x_i} \in R^d$ of a dataset D, only two class labels are produced $y_i \in \{+1, -1\}$. The hyperplane of x points is defined by the hyperplane function as shown in eq. 1 [3]:

$h(x) = 0$ where

$$h(x) = wTx + b = w1x1 + w2x2 + ...wdxd + b \quad (1)$$

Points in the hyperplane satisfy the eq. (2):

$$h(x) = wTx + b = 0 \quad (2)$$

Hyperplane separates the d-dimensional space in two half spaces. For a linearly separated dataset the class y for any point x will be:

$$y = \begin{cases} +1 \ if \ h(\boldsymbol{x}) > 0 \\ -1 \ if \ h(\boldsymbol{x}) < 0 \end{cases} \quad (3)$$

Assuming the xp projection of point x in the hyperplane, the directed distance is shown in eq. (4):

$$\boldsymbol{x} = \boldsymbol{x}_p + \boldsymbol{r}, \text{ and } \quad \boldsymbol{x} = \boldsymbol{x}_p + r\frac{\boldsymbol{w}}{\|\boldsymbol{w}\|} \quad (4)$$

where $\frac{w}{\|w\|}$ is the unit vector [3][5].

Using eq. (2) and (4) the directed distance of a point from the hyperplane is:

$$r = \frac{h(\boldsymbol{x})}{\|\boldsymbol{w}\|} \quad (5)$$

ISSN No: 2456-2156

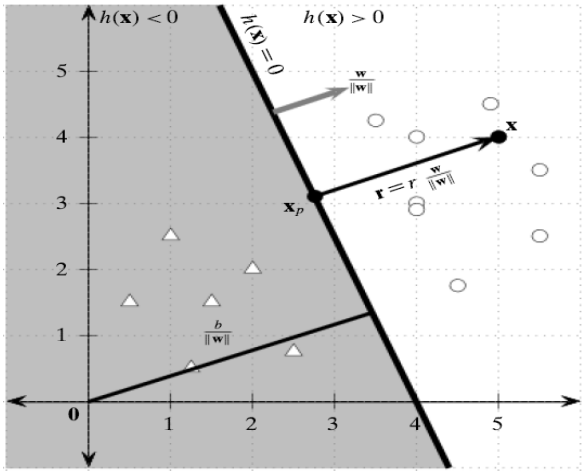

Figure 1 Geometry of a separating hyperplane in 2D

The distance of a point from the hyperplane h(x) = 0 is shown in eq. (6):

$$\delta = y\, r = \frac{y\, h(x)}{\|w\|} \qquad (6)$$

The margin is defined as the minimum distance of a point from the separating hyperplane, as shown in eq. (7):

$$\delta^* = \min_{x_i} \left\{ \frac{y_i (w^T x_i + b)}{\|w\|} \right\} \qquad (7)$$

Support vectors are points which lies in the margin of the classifier satisfying the relation $r = \delta^*$.

A. SVM linear case

In linearly separable hyperplanes, points are separated perfectly by the hyperplane. It is important to find the maximum margin separating hyperplane as the optimal one. According to eq. (7) the optimal hyperplane is found as:

$$h^* = arg\ \max_{h} \left\{ \frac{1}{w} \right\} \qquad (8)$$

Equivalently, it is needed to minimize the following relations:

Objective function: $\min_{w,b} \left\{ \frac{\|w\|^2}{2} \right\}$ (9)

And Linear Constraints:

$$y_i\ (w^T\ x_i + b) \geq 1\ for\ every\ x_i\ \in D$$

The dual problem is solving using Lagrange multipliers introducing a parameter αi for each constant satisfying the Karush-Kuhn-Tucker (KKT) condition to obtain the optimal solution:

$\alpha i\ (yi(wTxi + b)\ -1 = 0$ and $\alpha i \geq 0$ (10)

For the optimal hyperplane function h(x) = wT + b, the predicted class for the point z will be:

$$\hat{y} = sign\big(h(z)\big) = sign(w^T z + b) \qquad (11)$$

Which returns only +1 or -1 depending from the sign of the function.

B. Multilabel SVM

In practice, however, we often have to tackle problems involving K > 2 classes. Various methods have been proposed for combining multiple two-class SVMs in order to build a multiclass classifier. One commonly used approach is to construct K separate SVMs, in which the kth model yk(x) is trained using the data from class Ck as the positive examples and the data from the remaining K − 1 classes as the negative examples. This is known as the one-versus-the-rest approach. A different approach to multiclass classification, based on error-correcting output codes, was developed by [8] and applied to support vector machines by [9]. This can be viewed as a generalization of the voting scheme of the one-versus-one approach in which more general partitions of the classes are used to train the individual classifiers. The K classes themselves are represented as particular sets of responses from the two-class classifiers chosen, and together with a suitable decoding scheme, this gives robustness to errors and to ambiguity in the outputs of the individual classifiers. Although the application of SVMs to multiclass classification problems remains an open issue, in practice the one-versus-the-rest approach is the most widely used in spite of its ad-hoc formulation and its practical limitations [10].

C. Kernel SVM

Linear SVM approach can be used for datasets with a nonlinear decision boundary using the kernel. The d-dimensional points xi are mapped in φ(xi) points of a high-dimensional feature. For the given extra flexibility points φ(xi) are more likely to be linearly separable in the feature space. Kernel allows carrying operations in input space rather than in mapping points into feature space. To apply the kernel trick for nonlinear SVM classification, all operations require only the kernel function, as shown in eq. (12):

$K(xi, xj) = \varphi(xi)T\varphi(xj)$ (12)

Assuming that n points $\boldsymbol{x_i} \in D$ have respective labels yi, applying φ to each point a new dataset is obtained Dφ, in the feature space comprising the transformed points φ(xi) along their labels yi.

The SVM objective function in the feature space is shown in eq. (13):

Objective Function: $\min_{w,b,\xi_i} \left\{ \frac{\|w\|^2}{2} + C \sum_{i=1}^{n} (\xi_i)^2 \right\}$

Linear Constraints: $yi\ (wT\ \varphi(xi) + b \geq 1 - \xi i$ for $\xi i \geq 0$ (13)

Kernel are a group of mathematical functions used in SVM algorithms, requiring data as input and transform them in a desired form.

SVM kernels functions helps in changing the data dimensions. Kernels are used to analyze patterns in a dataset by solving non-liner problems using linear classifiers. SVM algorithms use kernel-trick to transform data-points and create

ISSN No: 2456-2156

optimal decision boundary. Kernel functions return a scalar product between two points in a suitable feature space as shown in eq. (14):

$$K(\bar{x}) = \begin{cases} 1 \; if \; \|\bar{x}\| \leq 1 \\ 0 \quad otherwise \end{cases} \qquad (14)$$

D. Linear kernel

It is the most basic type of kernel, usually one dimensional and with a lot of features and very faster. It is used for classification problems that can be linearly separated. Its formula is shown in eq. (15)

$$F(x, xj) = sum(x.xj) \qquad (15)$$

where x, xj are data points needed to be classified and '.' Is the dot product of both values.

Polynomial kernel

It is a generalization of linear kernel but less efficient and accurate. Its formula is shown in eq. (16):

$$F(x, xj) = (x.xj + 1)\,\hat{}\,d \qquad (16)$$

where '^' denotes the degree and F(x, xj) represents the decision boundary to separate the given classes.

E. Gaussian Radial Basis Function (RBF)

It is usually used for non-linear data, for separation where there is no prior knowledge of data. Its formula is shown in eq. (17):

$$F(x, xj) = exp(-gamma * ||x - xj||\hat{}2) \qquad (17)$$

Where the value of gamma varies from 0 to 1.

F. Sigmoid kernel

It is widely used in neural networks, similar with a two layer perceptron model which works as an activation function for neurons. Its formula is shown in eq. (18):

$$F(x, xj) = tanh(\alpha xay + c) \qquad (18)$$

G. Scaling the data with Standard Scaler Function

Scaling of features is an essential step in modeling algorithms with the datasets. The data, usually obtained from questionnaires, surveys, research, scraping, etc., contains features of various dimensions and scales leading to a biased outcome of prediction in terms of misclassifications error and accuracy rates. Scaling of data is necessary prior to data modeling.

Standardization of dataset is very important in learning algorithms. Outliers in the data set are removed using robust scalers, normalizers or transformers. Dataset may behave badly if their features are not standard normally distributed. Many elements in the objective function of a learning algorithm assume that all features are normally distributed, zero mean (centered around zero) and a unit variance.

Standardization scales data by converting the statistical distribution of the data with mean zero and unit standard deviation. Python sklearn library uses StandardScaler() function to transform the dataset in the standard format. An object from StandardScaler() function is created and fit_transform() function is used to with the object to transform data and standardize it as shown in eq. (19) [8]:

$$\text{Standardization: } z = \frac{x - \mu}{\sigma} \qquad (19)$$

where $mean\ \mu = \frac{1}{N}\sum_{i=1}^{N}(x_i)$ and $standard\ deviation\ \sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2}$

III. EXPERIMENTAL RESULTS

A. Data Exploration and Pre-Processing

The dataset consists in 3797 records containing the purchases for the first quarter of 2020 of the registered customers. The predicting attributes will be 6: Gender, Age, Value (total quarterly purchases), Category (Preferred Product Category), Quantity and Previous Year (total purchases of previous year for recurring clients and 0 for new clients). The clients are classified in 3 different labels [1,2,3] according to the total annual purchase for 2020. The aim is to predict the label for other clients based on the quarterly data. It is seen from fig. 2 that there is a slight class imbalance in the dataset, with the prevalence of class 1.

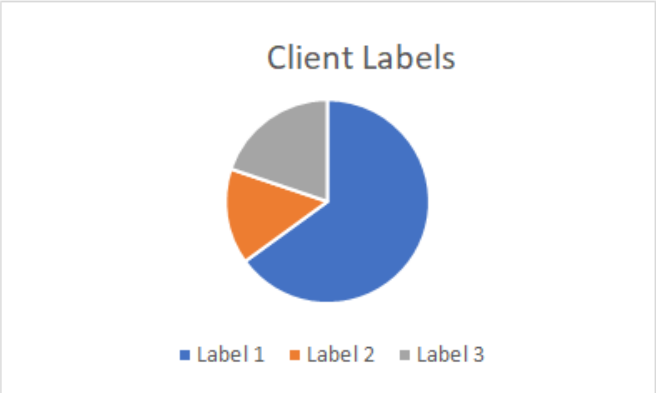

Figure 2 Client Labels in dataset

Before applying the model, some preprocessing is done on the data. We print a summary information of our dataset in order to understand the missing values or other irregularities if any.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3797 entries, 0 to 3796
Data columns (total 9 columns):
 #   Column   Non-Null Count  Dtype
---  ------   --------------  -----
 0   CLIENTE  3797 non-null   int64
 1   GENDER   3797 non-null   object
 2   CITY     3797 non-null   object
 3   CAT      3797 non-null   int64
 4   AGE      2878 non-null   float64
 5   Val      3797 non-null   int64
```

```
 6  Qty     3797 non-null   int64
 7  Prev    3002 non-null   int64
 8  Label   3797 non-null   int64
dtypes: float64(1), int64(6), object(2)
memory usage: 267.1+ KB
```

It can be observed that the age is missing for some of the clients and also for the new clients, there is a null value in the Previous_Year column. The median age is imputed in the Age column for the missing records and 0 is imputed in the missing Previous_Year column as it belongs to new clients. Furthermore the Sex column values are encoded so they can be transformed in numerical values. Furthermore, we don't use the city value as a predictor as it is the same for more than 95% of the data.

B. Fitting the model to the dataset

After the pre-processing step, the SVM model is applied to the dataset. The model is applied firstly without scaling the data to emphasize the importance of data standardization in SVM applications. The data is divided in 80% used for training the algorithm and 20% for test purposes. The model is fitted to the training data. SVM is used with a with a linear kernel first and the accuracy is recorded.

From implementations in python results that the run time is extremely affected and it took approximately 29 minutes to fit the data. In the non-scaled data implementation we have an accuracy of 76.71%.

Next, the data is scaled using the Standard Scaler Function and the algorithm is reapplied. The accuracy obtained is 80.79 % and the run time is almost instantaneous.

TABLE I. ACCURACY AND RUN-TIME RESULTS

	Accuracy	**Run-Time**
Not Standardized data	76.71%	28min 38s
Standardized data	80.79%	250 ms

In order to validate our scores we perform cross validation on the model with 10 folds. The results are described in table 2 and it can be seen they are pretty consistent with our value, so the model is validated.

TABLE II. CROSS VALIDATION RESULTS

Fold	Accuracy
Fold 1	0.7730263
Fold 2	0.7796053
Fold 3	0.8026316
Fold 4	0.8026316
Fold 5	0.7960526
Fold 6	0.7960526
Fold 7	0.7927632
Fold 8	0.7788779
Fold 9	0.7821782
Fold 10	0.7887789
Average	0.7892598

After validating our model we check the confusion table to understand how the model performs for each individual label.

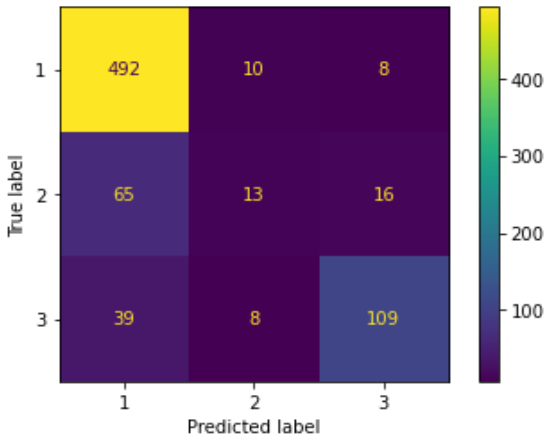

Figure 3 Confusion Matrix SVM

As it can also be expected, our model works fine on the first and third label but suffers on the middle label. We are able to predict the middle class 2 correctly 13 times. We are doing a very good job in predicting the first and third label predicting the correct label with a precision score of approximately 82%. The accuracy is measured for different type of kernels and the results are displayed in figure 4.

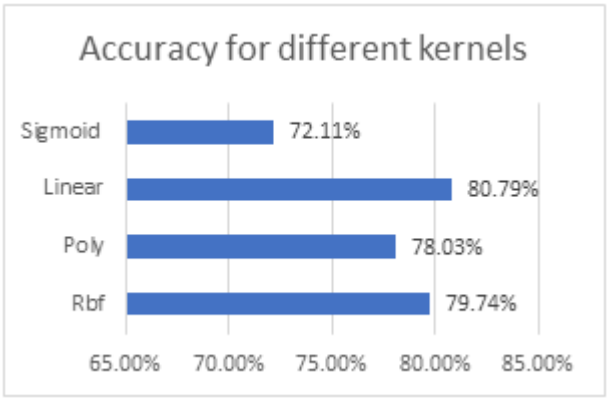

Figure 4 Accuracy for different type of kernels

It is observed that best results are obtained using the linear kernel for our dataset since it produces the best accuracy value.

IV. CONCLUSIONS

After the text edit has been completed, the paper is ready for the template. Duplicate the template file by using the Save As command, and use the naming convention prescribed by your conference for the name of your paper. In this newly created file, highlight all of the contents and import your prepared text file. You are now ready to style your paper; use the scroll down window on the left of the MS Word Formatting toolbar.

In this paper, we build a model based on SVM (Support Vector Machines) to predict customer purchases. Although the size of the dataset is limited, our prediction accuracy is good and it can be applied in every retail dataset as we used real production data. Our model combines client attributes and preferences with historic purchase data as predictors. It is observed that data scaling is critical before applying SVM, resulting in extremely shorter run time and better accuracy. The proposed model was tested for different type of kernels and got the best results for the linear kernel. We observed some difficulties in predicting the middle label 2, mostly due to the smaller number of clients in that class and class imbalance.

FUTURE WORK

In the future, the study will be carried for larger datasets and will be expanded in different dimensions of the data. Also, techniques like up sampling or SMOTE will be performed to reduce the effect of class imbalance.

REFERENCES

[1]. Tractica, Artificial Intelligence for Enterprise Applications, 2020
[2]. A Support Vector Machine Approach for Churn Prediction in Telecom Industry August 2014, International Journal on Information 17
[3]. Mohammed J. Zaki, Wagner Meira Jr., Data Mining and Machine Learning Fundamental Concepts and Algorithms, Cambridge University Press, 2020
[4]. K. Maheswari and P. P. A. Priya, "Predicting customer behavior in online shopping using SVM classifier," 2017 IEEE International Conference on Intelligent Techniques in Control, Optimization and Signal Processing (INCOS), 2017, pp. 1-5, doi: 10.1109/ITCOSP.2017.8303085.
[5]. He Z., Cichocki A. (2007) An Efficient K-Hyperplane Clustering Algorithm and Its Application to Sparse Component Analysis. In: Liu D., Fei S., Hou Z., Zhang H., Sun C. (eds) Advances in Neural Networks – ISNN 2007. ISNN 2007. Lecture Notes in Computer Science, vol 4492. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-72393-6_122
[6]. Zhao Y., Li B., Li X., Liu W., Ren S. (2005) Customer Churn Prediction Using Improved One-Class Support Vector Machine. In: Li X., Wang S., Dong Z.Y. (eds) Advanced Data Mining and Applications. ADMA 2005. Lecture Notes in Computer Science, vol 3584. Springer, Berlin, Heidelberg. https://doi.org/10.1007/11527503_36
[7]. Thomas G. Dietterich, Ghulum Bakiri, (1995). Solving Multiclass Learning Problems via Error-Correcting Output Codes. Computer Science. DOI: https://doi.org/10.1613/jair.105
[8]. Allwein, E. L., Schapire, R. E., & Singer, Y. (2001). Reducing multiclass to binary: A unifying approach for margin classifiers. Journal of Machine Learning Research, 1(2), 113-141.
[9]. https://scikit-learn.org/stable/modules/preprocessing.html
[10]. Christopher M. Bishop. 2006. Pattern Recognition and Machine Learning (Information Science and Statistics). Springer-Verlag, Berlin, Heidelberg. 2006