Binary Classification of the Iranian churn dataset using learning models

Leonel Guerrero
Carnet: 18-10638

April 19, 2023

Description

In this document we will use the Iranian churn dataset to predict the churn of the customers. the dataset is available in the UCI Machine Learning Repository.

Information about the dataset

This dataset is randomly collected from an Iranian telecom company database over a period of 12 months. A total of 3150 rows of data, each representing a customer, bear information for 13 columns. The attributes that are in this dataset are call failures, frequency of SMS, number of complaints, number of distinct calls, subscription length, age group, the charge amount, type of service, seconds of use, status, frequency of use, and Customer Value.

All of the attributes except for attribute churn is the aggregated data of the first 9 months. The churn labels are the state of the customers at the end of 12 months. The three months is the designated planning gap.

Attribute Information

- Anonymous Customer ID
- Call Failures: number of call failures
- Complains: binary (0: No complaint, 1: complaint)
- Subscription Length: total months of subscription
- Charge Amount: Ordinal attribute (0: lowest amount, 9: highest - amount)
- Seconds of Use: total seconds of calls
- Frequency of use: total number of calls
- Frequency of SMS: total number of text messages
- Distinct Called Numbers: total number of distinct phone calls
- Age Group: ordinal attribute (1: younger age, 5: older age)
- Tariff Plan: binary (1: Pay as you go, 2: contractual)

- Status: binary (1: active, 2: non-active)
- Churn: binary (1: churn, 0: non-churn) - Class label
- Customer Value: The calculated value of customer

Data Analysis

Show the raw data

First, we will analyze the dataset and then we will use different learning models to predict the churn of the customers. Let's start by showing the first and last 5 rows of the dataset.

	1.0	2.0	3.0	4.0	5.0
Call Failure	8.0	0.0	10.0	10.0	3.0
Complains	0.0	0.0	0.0	0.0	0.0
Subscription Length	38.0	39.0	37.0	38.0	38.0
Charge Amount	0.0	0.0	0.0	0.0	0.0
Seconds of Use	4370.0	318.0	2453.0	4198.0	2393.0
Frequency of use	71.0	5.0	60.0	66.0	58.0
Frequency of SMS	5.0	7.0	359.0	1.0	2.0
Distinct Called Numbers	17.0	4.0	24.0	35.0	33.0
Age Group	3.0	2.0	3.0	1.0	1.0
Tariff Plan	1.0	1.0	1.0	1.0	1.0
Status	1.0	2.0	1.0	1.0	1.0
Churn	0.0	0.0	0.0	0.0	0.0
Customer Value	132.6	17.46	181.29	252.48	144.78

Table 1: First 5 rows of the dataset

	3146.0	3147.0	3148.0	3149.0	3150.0
Call Failure	21.0	17.0	13.0	7.0	8.0
Complains	0.0	0.0	0.0	0.0	1.0
Subscription Length	19.0	17.0	18.0	11.0	11.0
Charge Amount	2.0	1.0	4.0	2.0	2.0
Seconds of Use	6697.0	9237.0	3157.0	4695.0	1792.0
Frequency of use	147.0	177.0	51.0	46.0	25.0
Frequency of SMS	92.0	80.0	38.0	222.0	7.0
Distinct Called Numbers	44.0	42.0	21.0	12.0	9.0
Age Group	2.0	5.0	3.0	3.0	3.0
Tariff Plan	2.0	1.0	1.0	1.0	1.0
Status	1.0	1.0	1.0	1.0	1.0
Churn	0.0	0.0	0.0	0.0	1.0
Customer Value	342.765	50.185	106.11	207.45	55.86

Table 2: Last 5 rows of the dataset

Now we will show the summary of the dataset.

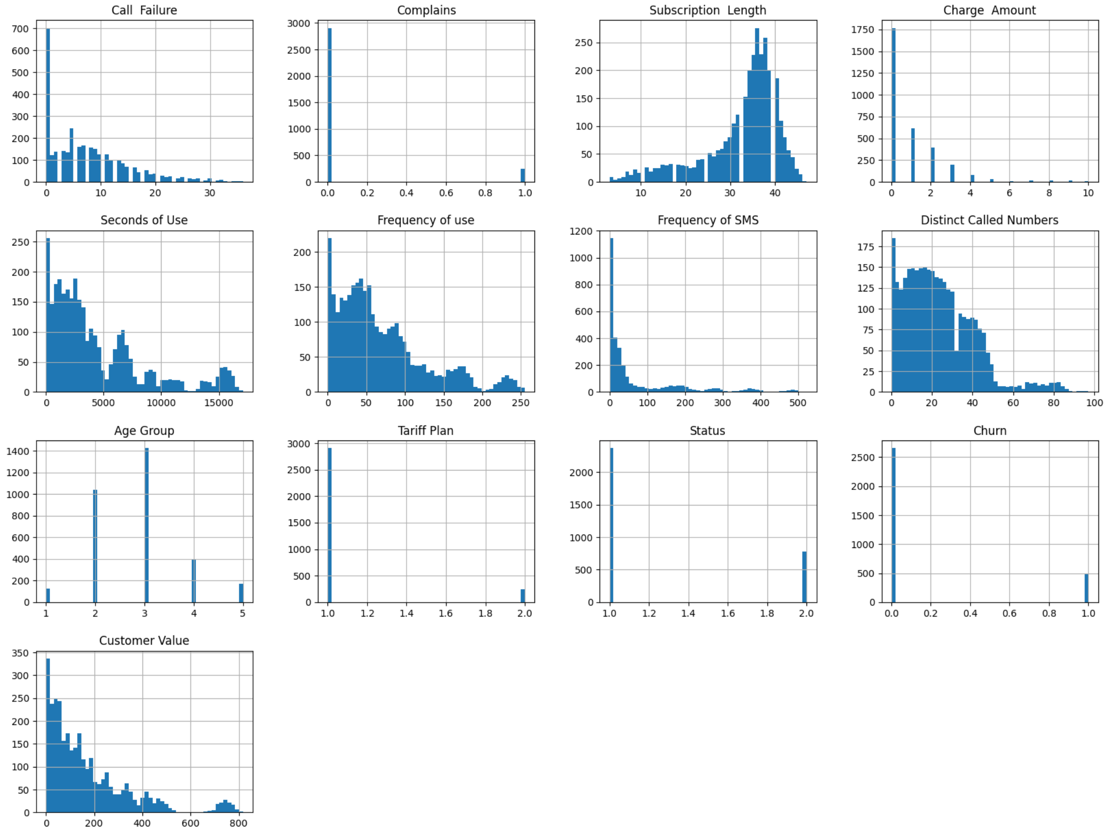


Figure 1: Histogram of the attributes

	mean	std	min	25

Table 3: Summary of the dataset

Data Visualization

Now we will show in the figure 1 the histograms of each attribute to see the distribution of the data, and to see if there are any outliers.

Analysis of the dataset

Through the analysis of the data and the graphs it can be appreciated that, in the binary characteristics there are many more data in one of the characteristics in the other, a fact that could affect the learning models, on the other hand it can also be appreciated that the distribution of the data, with the exception of the age group does not have a normal distribution, and moreover, most of them tend to group the data in the initial values tending to an inverse distribution, with the exception of the subscription length.

Feature scaling

Due to the analysis performed on the data and the fact that some learning algorithms that will be used use the gradient descent, which is sensitive to the scale of the data, the decision was made to perform a scaling of the data to improve the convergence processes of the learning algorithms, a normalization, a standardization and the combination of the two will be performed to train each of the models and compare the results obtained and get the best model that manages to classify the data.

Splitting the dataset

Apart from the scaling of the data, the data set will be separated in a proportion of 80% and 20% where the same 80% of the data will be used to train each of the learning machines and the effectiveness of each of the machines will be verified with the remaining 20% of the data.

Learning models

The learning machines that will be used to try to classify the data will be the following:

- Perceptron: to know and test if the data are linearly separable.
- MLP: In case the data are not linearly separable, look for a more complex separation rule by the use of the gradient descent algorithm, in the same way this learning machine will be used due to its versatility for the separability of more complex regions.
- SVM: This machine will be used to classify the data taking a different approach to the previous learning machine in order to compare different algorithms and see which one classifies the data better.

Perceptron

The first learning machine that will be used is the perceptron, this machine is a linear classifier, which means that it will try to find a linear separation rule between the data. After training the machine, the results obtained are shown in the table **??**.

	Train		Test	
	Score	Errors	Score	Errors
Original	0.8476	384	0.8429	99
Standardized	0.8159	464	0.8175	115
Normalized	0.854	368	0.8429	99
Standardized and Normalized	0.8671	335	0.8571	90

Through the table we can see that the machine was able to classify the data with an accuracy between 81.59% and 86.71%, in the training phase and between 81.75% and 85.71% in the test phase, the best results were obtained with the combination of the normalization and standardization of the data, with the highest accuracy in the test phase, and the lowest number of errors.

MLP

The second learning machine that will be used is the MLP, this machine is a non-linear classifier, which means that it will try to find a non-linear separation rule between the data. After training the machine, the results obtained are shown in the table 5.

Tuning the parameters of the MLP

The MLP is a learning machine that has a large number of parameters that can be tuned to improve the performance of the machine, in this case the parameters that will be tuned are the following:

- Number of neurons in the hidden layer: The number of neurons that will be used will be 3, 5 and 7
- Activation function: The activation functions that will be used will be the sigmoid, the tanh and the relu.

Results of the MLP

Due to the large number of combinations that can be made with the parameters of the MLP, we will only show the results of the best combinations of the parameters, specifically the best 10 combinations of the parameters, after training the machine, the results obtained are shown in the table 5 in the order of the best results to the worst.

			Train		**Test**	
Hidden Layer Sizes	**Activation**	**Dataset Type**	**Score**	**Errors**	**Score**	**Errors**
7	**relu**	**Standardized and Normalized**	0.902	247	0.8889	70
5	**relu**	**Standardized and Normalized**	0.902	247	0.8889	70
7	**relu**	**Standardized**	0.9036	243	0.8889	70
	logistic	**Standardized**	0.9	252	0.8889	70
	tanh	**Standardized and Normalized**	0.902	247	0.8889	70
5	**logistic**	**Standardized**	0.9004	251	0.8889	70
	relu	**Standardized**	0.9008	250	0.8857	72
3	**tanh**	**Standardized and Normalized**	0.898	257	0.8857	72
		Standardized	0.9008	250	0.8841	73
	relu	**Standardized and Normalized**	0.8917	273	0.8825	74

Table 5: Results of the MLP

Through the table we can see that from the 10 best combinations of the parameters, 7 of them are the same, with the same accuracy in the test phase, and the same number of errors, with 88.89% accuracy and 70 errors. Something worth mentioning is the fact that in the best machines neither the original dataset nor the normalized dataset is displayed, apart from the fact that the most repeated activation function is relu. Therefore, it can be seen that the combination of the normalization and standardization of the data, with the relu activation function, is a good combination of parameters for the MLP.

SVM

The third learning machine that will be used is the SVM, this machine is a non-linear classifier, which means that it will try to find a non-linear separation rule between the data. After training the machine, the results obtained are shown in the table 6.

Tuning the parameters of the SVM

The SVM is a learning machine that has a large number of parameters that can be tuned to improve the performance of the machine, in this case the parameters that will be tuned are the following:

- Kernel: The kernels that will be used will be the linear, the polynomial and the rbf.
- C: The C value represents the penalty of the error term (regularization), the values that will be used will be 0.1, 1 and 10.
- Gamma/degree: The gamma value represents the influence of a single training example, the values that will be used will be the calculate by the library, that are 'scale' (1 / (n_features * X.var())) and 'auto' (1 / n_features). In the case of the polynomial kernel, the max degree used by the polynomial, the values that will be used will be 2, 3 and 4.

Results of the SVM

Due to the large number of combinations that can be made with the parameters of the SVM, we will only show the results of the best combinations of the parameters, specifically the best 10 combinations of the parameters, after training the machine, the results obtained are shown in the table 6 in the order of the best results to the worst.

				Train		Test	
				Score	Errors	Score	Errors
Kernel	**C**	**Gamma/Degree**	**Dataset Type**				
Polynomial	**10**	**4**	**Standardized and Normalized**	0.9655	87	0.9476	33
RBF	**10**	**scale**	**Standardized and Normalized**	0.9631	93	0.9413	37
Polynomial	**10**	**3**	**Standardized and Normalized**	0.9627	94	0.9413	37
RBF	**10**	**auto**	**Standardized**	0.9579	106	0.9381	39
		scale	**Standardized**	0.9579	106	0.9381	39
Polynomial	**10**	**4**	**Standardized**	0.9512	123	0.9349	41
		3	**Standardized**	0.9532	118	0.9317	43
		2	**Standardized and Normalized**	0.9345	165	0.9286	45
	1	**3**	**Standardized and Normalized**	0.9377	157	0.9238	48

		4	**Standardized and Normalized**	0.9417	147	0.9238	48

Table 6: Results of the SVM

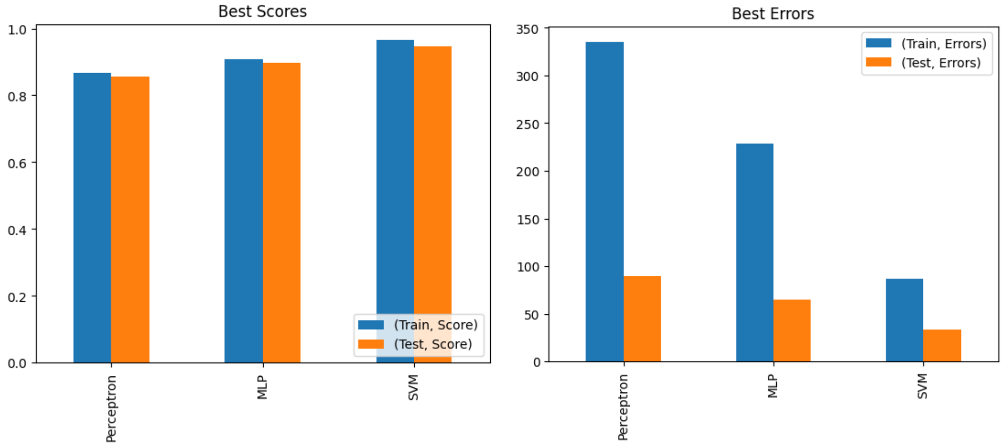
Through the table we can see that the best machines have similar results, with scores between 92.38% and 94.76%, also similar to MLP, in the best machines don't appear neither the original dataset nor the only normalized dataset. On the other hand it can be seen that in almost all of the best machines the best value of c was 10, where the best machine was the one with the polynomial kernel, $c = 10$, degree $= 4$ and a dataset that was standardized and normalized, with a score of 94.76% in the test set and 96.55% in the training set, and only 33 errors in the test set.

Comparison of the three models

Now we will compare the best machine of each model. A comparison table of the three models can be seen in the table 7.

	Train		**Test**		**Dataset type**
	Score	Errors	Score	Errors	
Perceptron	0.8671	335	0.8571	90	Standardized and Normalized
MLP	0.9091	229	0.8968	65	Standardized and Normalized
SVM	0.9655	87	0.9476	33	Standardized and Normalized

Table 7: Comparison of the three models



(a) Scores of the best machine of each model

(b) Errors of the best machine of each model

Figure 2: Comparison of the three models

Through the table and the figures we can see that the best machine of each model has a scores higher that 85% and a number of errors lower than 100, and the best machine is the SVM with a score of 94.76% and only 33 errors.

Code and complete results

The code of the project can be found in the following links:

- Github
- Colab