Churn prediction in telecommunication industry using kernel Support Vector Machines

Sayyam Anil Gada

ITCS- 5156 Fall 2022

October 12, 2022

This report is presented as a survey of a previous work [1]. Any assertions made within are subjective and do not represent those of the original author.

1. Introduction

In the Telecommunication Industry, customer churn detection is one of the most important research topics that the company must deal with retaining on-hand customers. Churn means the loss of customers due to existing offers of the competitors or maybe due to network issues.

1.1 Problem Statement

The problem of churn prediction is to split the following tasks:

- Input data analysis and pre-processing.
- Building a model to classify whether a customer will stay or churn.
- Evaluation of model concerning chosen metrics.

1.2 Motivation

- In this age of fierce competitions, customer retention is one of the most important tasks for many companies.
- Churn rate has a substantial impact on the lifetime value of the customer because it affects the future revenue of the company and the length of service.
- Due to a direct effect on the income of the industry, the companies are looking for a model that can predict customer churn.
- In this project, I will create a classification machine learning model which will predict whether a customer will be retained or churned.

2. Related Works

2.1 Churn Prediction of Customer in Telecom Industry using Machine Learning Algorithms

In May 2020 at the International Journal of Engineering Research & Technology (IJERT), V.Kavitha et al [2] presented their work – Churn Prediction in Telecom Industry using various machine learning algorithms.

They proposed a system in which they used various algorithms like Random Forest, XGBoost and Logistic Regression on a dataset which is different from Nguyen Nhu Y et al [1].

The work stated that they did data filtering, noise removal and feature selection as a part of data pre-processing, but they haven't given any information on what data they did data pre-processing and using which methods in the process.

The work then explained about the three classification methods to be used- Random Forest, Logistic Regression and XGBoost.

The data is sourced from Kaggle, and Training-Testing data is split into 80:20.

	precision	recall	f1-score	support
0	0.62	0.52	0.56	440
1	0.85	0.89	0.87	1321
accuracy			0.80	1761
macro avg	0.73	0.70	0.72	1761
weighted avg	0.79	0.80	0.79	1761

Fig 3.Confusion matrix of Random Forest.

	precision	recall	f1-score	support
0	0.57	0.56	0.56	440
1	0.85	0.86	0.86	1321
accuracy			0.79	1761
macro avg	0.71	0.71	0.71	1761
weighted avg	0.78	0.79	0.78	1761

Fig 4.Confusion matrix of Logistic regression.

	precision	recall	f1-score	support
0	0.58	0.50	0.54	440
1	0.84	0.88	0.86	1321
accuracy			0.78	1761
macro avg	0.71	0.69	0.70	1761
weighted avg	0.78	0.78	0.78	1761

Fig 5.Confusion matrix of XGBoost.

The results of the tables indicate that Random Forest works better for this dataset.

2.2 Churn Prediction Using Machine Learning and Recommendations Plans for Telecoms

In November 2019 at the Journal of Computer and Communications, Khulood Ebrah et al [3] presented their work about Churn Prediction Using Machine Learning and Recommendations Plans for Telecoms.

There are three machine learning models used in this work- Naïve Bayes, Support Vector Machine (SVM) and Decision Tree.

The algorithms are tested on two different datasets- IBM Watson dataset and cell2cell dataset.

During pre-processing step, the authors removed unnecessary features and extracted the features using ranking.

Principal Component Analysis (PCA) is used here for feature extraction and data is split into 70% training and 30% testing.

The model’s performance has been measured by area under curve where the best AUCs are (0.82, 0.87, 0.78) for IBM dataset & (0.98, 0.99, 0.98) for cell2cell dataset. The AUC, which obtained using SVM algorithm, is better compared with the previous papers.

3. Methods

3.1 Support Vector Machines (SVM)

Support Vector Machines [4] are a type of supervised machine learning algorithm that provides analysis of data for classification and regression analysis. While they can be used for regression, SVM is mostly used for classification. We carry out plotting in the n-dimensional space. Value of each feature is also the value of the specific coordinate. Then, we find the ideal hyperplane that differentiates between the two classes. These support vectors are the coordinate representations of individual observation. It is a frontier method for segregating the two classes.

3.2 Linear Kernel

It is the most basic type of kernel [5], usually one dimensional in nature. It proves to be the best function when there are lots of features. The linear kernel is mostly preferred for text-classification problems as most of these kinds of classification problems can be linearly separated.

Linear kernel functions are faster than other functions.

Linear Kernel Formula

F(x, xj) = sum(x.xj)

Here, x, xj represents the data you're trying to classify.

3.3 Polynomial Kernel

It is a more generalized representation of the linear kernel. It is not as preferred as other kernel functions as it is less efficient and accurate.

Polynomial Kernel Formula

F(x, xj) = (x.xj+1)^d

Here '.' shows the dot product of both the values, and d denotes the degree.

F(x, xj) representing the decision boundary to separate the given classes.

3.4 Gaussian Radial Basis Function (RBF)

It is one of the most preferred and used kernel functions in svm. It is usually chosen for non-linear data. It helps to make proper separation when there is no prior knowledge of data.

Gaussian Radial Basis Formula

F(x, xj) = exp(-gamma * ||x - xj||^2)

The value of gamma varies from 0 to 1. You must manually provide the value of gamma in the code. The most preferred value for gamma is 0.1.

3.5 Sigmoid Kernel

It is mostly preferred for neural networks. This kernel function is like a two-layer perceptron model of the neural network, which works as an activation function for neurons.

It can be shown as,

Sigmoid Kenel Function

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

4. Experiments

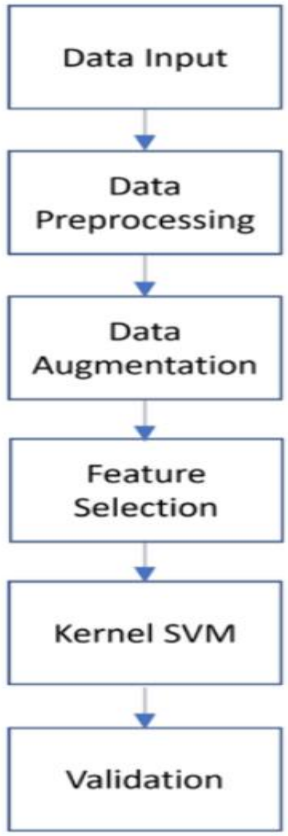

Fig: Implementation steps

4.1 Data Input

The train and test datasets are downloaded from Kaggle [6], and they are split into two different datasets maintaining a ratio of 80-20.

4.2 Data Preprocessing

During data pre-processing, Churn feature which has datatype of Boolean, is converted into int64 using Label Encoder.

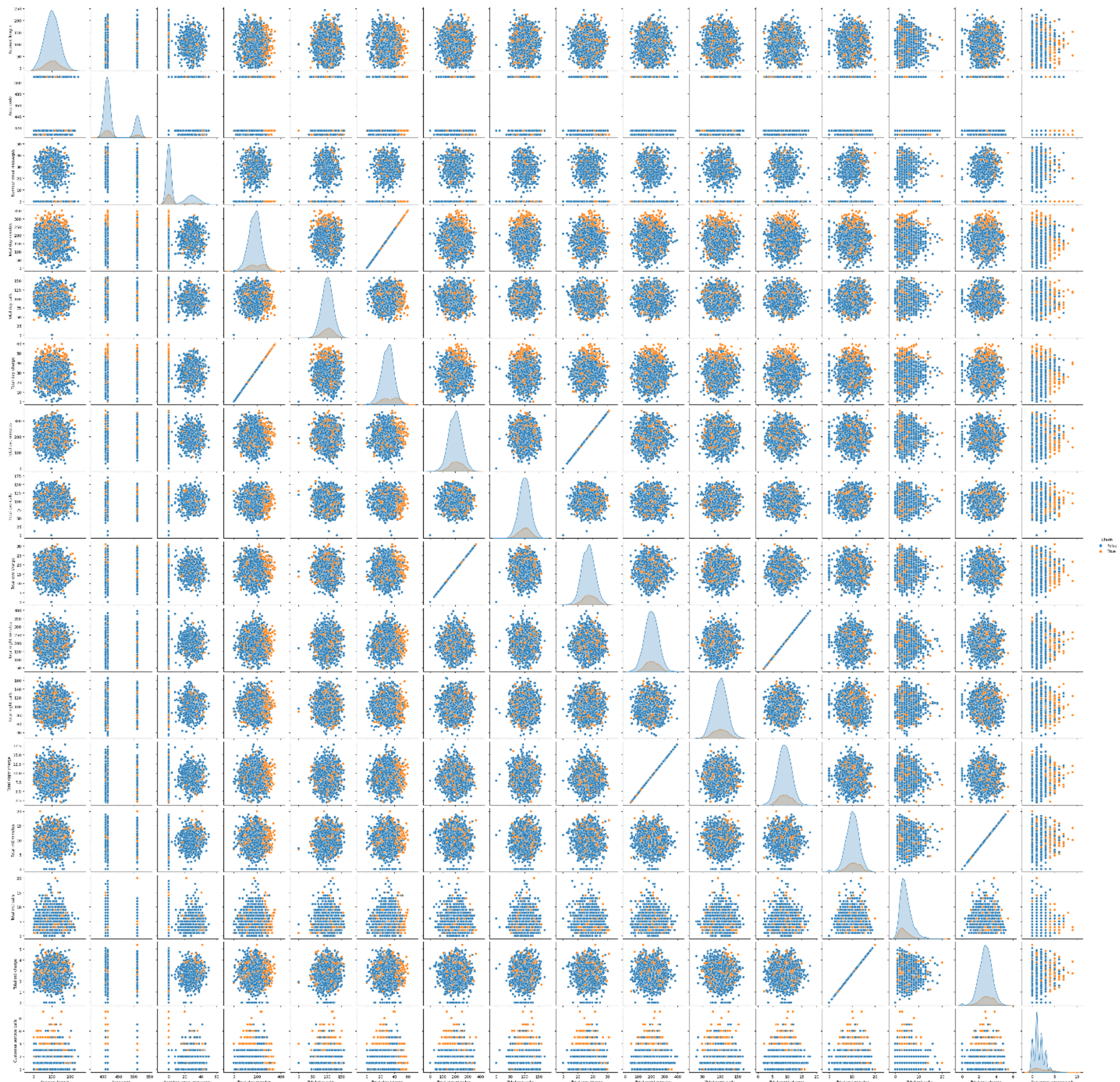

Fig: Pair plot of the dataset

While looking at pairplot graph, we came to know that- Total day minutes and Total day charge, Total eve minutes and Total eve charge, Total night minutes and Total night charge and lastly Total intl minutes and Total intl charge are correlated. So, we will only select one of them.

Along with them, state and area code were also removed as they contained null and insignificant data.

4.3 Data Augmentation

During data augmentation, data standardization was done using Z-score. Z-score makes data consistent by rescaling the values of variables to be centered around the mean of 0 with standard deviation of 1.

The formula for calculating a z-score is is $z = (x-\mu)/\sigma$, where x is the raw score, μ is the population mean, and σ is the population standard deviation.

4.4 Feature Selection

The paper suggested two methods for feature selection- Filter method using Pearson's Correlation and Wrapper method.

I used Wrapper method and did both the Sequential Forward Selection and Sequential Backward Selection using K-Nearest Neighbors (KNN) on the dataset and found out that total day minutes, total eve minutes and customer service calls are the most useful features in the dataset.

4.5 Kernel SVM & Validation

```
Train Accuracy: 0.794074
Test Accuracy: 0.794603
MCC: 0.017276718360329745
              precision    recall  f1-score   support

           0       0.86      0.91      0.88       572
           1       0.16      0.11      0.13        95

    accuracy                           0.79       667
   macro avg       0.51      0.51      0.51       667
weighted avg       0.76      0.79      0.78       667
```

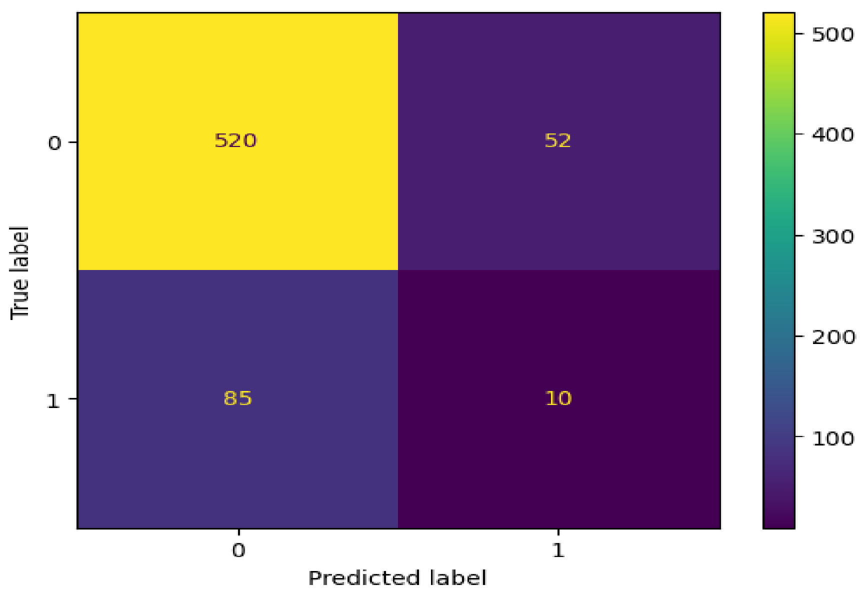

Fig: Sigmoid kernel Support Vector Machine without Feature Selection

```
Train Accuracy: 0.854464
Test Accuracy: 0.857571
MCC: 0.0
              precision    recall  f1-score   support

           0       0.86      1.00      0.92       572
           1       0.00      0.00      0.00        95

    accuracy                           0.86       667
   macro avg       0.43      0.50      0.46       667
weighted avg       0.74      0.86      0.79       667
```

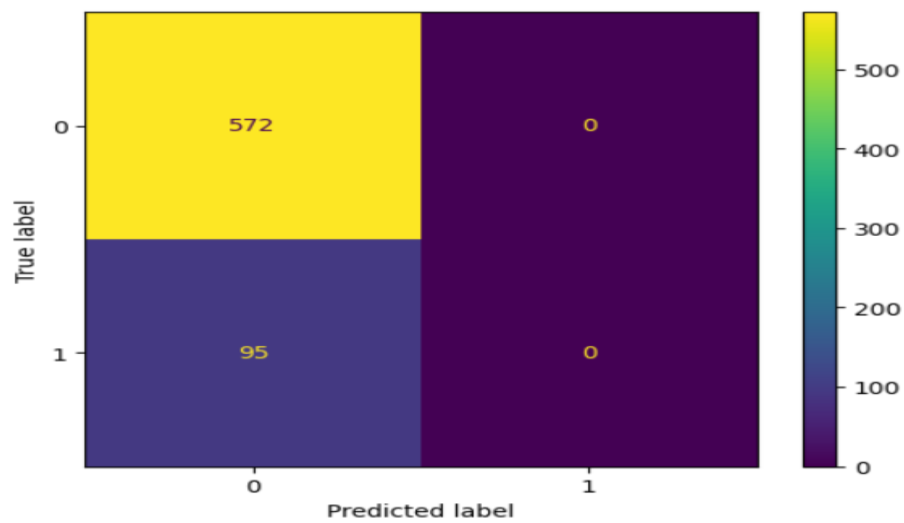

Fig: Linear kernel Support Vector Machine without Feature Selection

```
Train Accuracy: 0.943736
Test Accuracy: 0.920540
MCC: 0.6335091761022514
              precision    recall  f1-score   support

           0       0.92      0.99      0.96       572
           1       0.90      0.49      0.64        95

    accuracy                           0.92       667
   macro avg       0.91      0.74      0.80       667
weighted avg       0.92      0.92      0.91       667
```

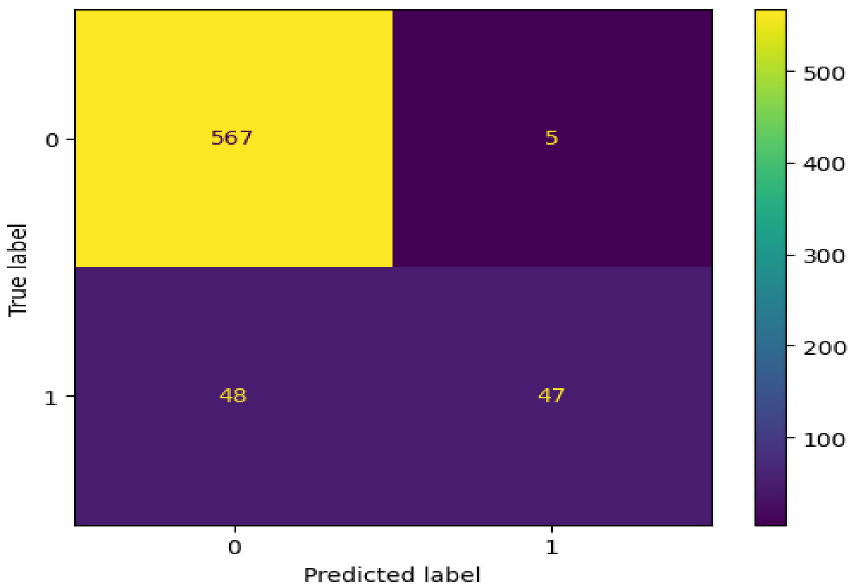

Fig: Rbf kernel Support Vector Machine without Feature Selection

```
Train Accuracy: 0.942236
Test Accuracy: 0.926537
MCC: 0.6653805094313474
              precision    recall  f1-score   support

           0       0.93      0.99      0.96       572
           1       0.93      0.53      0.67        95

    accuracy                           0.93       667
   macro avg       0.93      0.76      0.81       667
weighted avg       0.93      0.93      0.92       667
```

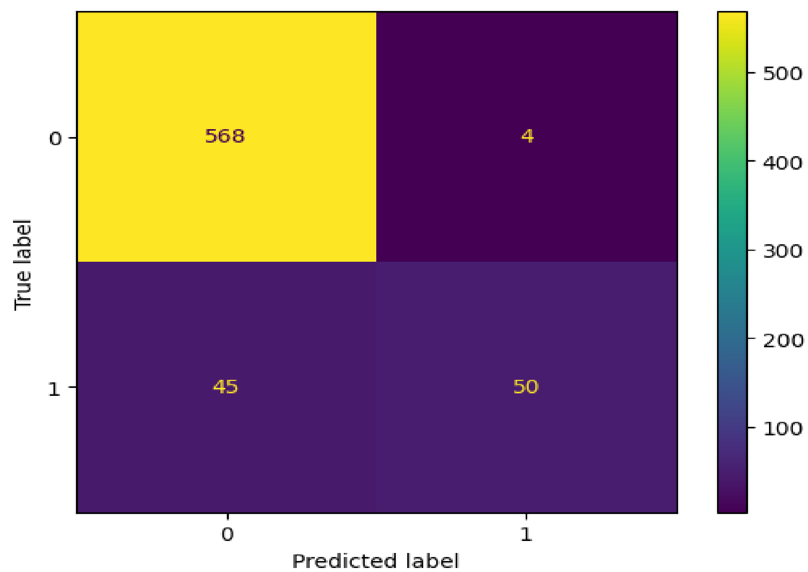

Fig: Polynomial kernel Support Vector Machine without Feature Selection

```
Train Accuracy: 0.887097
Test Accuracy: 0.902549
MCC: 0.5339028910261884
              precision    recall  f1-score   support

           0       0.91      0.98      0.95       572
           1       0.81      0.41      0.55        95

    accuracy                           0.90       667
   macro avg       0.86      0.70      0.75       667
weighted avg       0.90      0.90      0.89       667
```

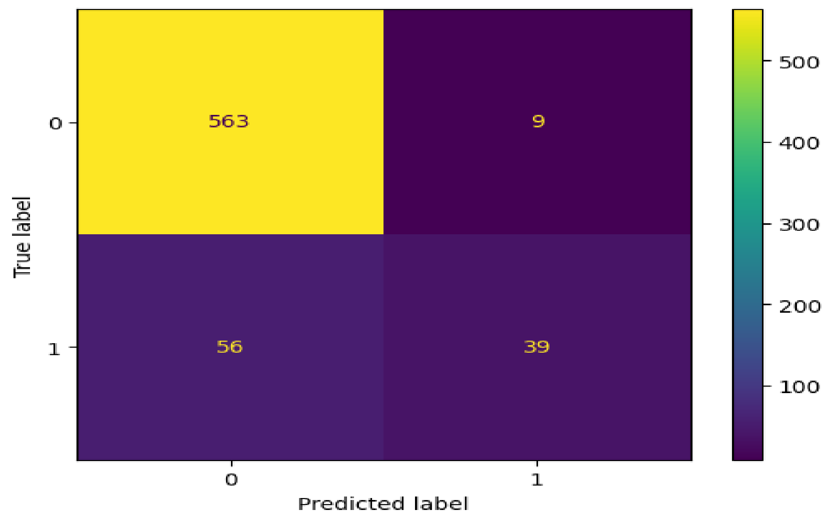

Fig: Polynomial kernel Support Vector Machine with Feature Selection

5. Conclusion

The Rbf kernel Support Vector Machine is the best algorithm for Training accuracy. Polynomial kernel Support Vector Machine is the best overall.

After comparing the machine learning predictions, I came to know that using only features which were shown as the most useful by the two wrapper methods for machine learning reduced the performance of the algorithm. This shows to me that less important features also make an impact to get better results.

The results I obtained are far different than those from the referred work. The work got rbf as the best kernel with accuracy of 98.9% whereas I got Polynomial as the best kernel with accuracy of 93% which indicates that I may have done data pre-processing differently.

I learned new concepts like converting Boolean feature into integer using Label Encoding and calculating the Z-score for data standardization. Data pre-processing is the most difficult part of the process as data needs to be cleaned, standardized, and converted according to the requirements.

6. References

[1] Nguyen Nhu Y, Tran Van Ly, Dao Vu Truong Son (2022), Churn prediction in telecommunication industry using kernel Support Vector Machines, Public Library of Science, https://doi.org/10.1371/journal.pone.0267935

[2] Kavitha, V. & Kumar, G. & Kumar, S. & Harish, M. (2020), Churn Prediction of Customer in Telecom Industry using Machine Learning Algorithms, International Journal of Engineering Research and. V9. 10.17577/IJERTV9IS050022.

[3] Ebrah K, Elnashra S (2019) Churn prediction using machine learning and recommendations plans for telecoms. Journal of Computer and Communications 7(11). Article Id 96177

[4] https://data-flair.training/blogs/svm-support-vector-machine-tutorial/

[5] https://dataaspirant.com/svm-kernels/

[6] https://www.kaggle.com/datasets/mnassrib/telecom-churn-datasets