Implementation of Machine Learning Algorithms for Customer Churn Prediction

Manal Loukili1, Fayçal Messaoudi2, Raouya El Youbi1

1. National School of Applied Sciences, Sidi Mohamed Ben Abdellah University, Fez, Morocco
2. National School of Business and Management, Sidi Mohamed Ben Abdellah University, Fez, Morocco

Received: 20 Feb 2022/ Revised: 04 Jul 2022/ Accepted: 10 Aug 2022

Abstract
Churn prediction is one of the most critical issues in the telecommunications industry. The possibilities of predicting churn have increased considerably due to the remarkable progress made in the field of machine learning and artificial intelligence. In this context, we propose the following process which consists of six stages. The first phase consists of data preprocessing, followed by feature analysis. In the third phase, the selection of features. Then the data was divided into two parts: the training set and the test set. In the prediction process, the most popular predictive models were adopted, namely random forest, k-nearest neighbor, and support vector machine. In addition, we used cross-validation on the training set for hyperparameter tuning and to avoid model overfitting. Then, the results obtained on the test set were evaluated using the confusion matrix and the AUC curve. Finally, we found that the models used gave high accuracy values (over 79%). The highest AUC score, 84%, is achieved by the SVM and bagging classifiers as an ensemble method which surpasses them.

Keywords: Machine Learning; Churn Prediction; Bagging SVM; k-NN; Random Forest.

1- Introduction
The exponential growth in the number of operators in the market, due to globalization and advances in the telecommunications industry, is increasing competition. In this competitive era, it has become imperative to maximize profits regularly, for this reason, various approaches have been proposed, especially acquiring new customers, up-selling existing customers and increasing the retention period of current customers. Retaining existing customers is the simplest and least expensive strategy compared to the others. In order to adopt this strategy, companies need to reduce the eventual churn of customers. The main reason for this loss of customers is dissatisfaction with the services provided to consumers and the support mechanism. To solve this problem, the solution is to predict which customers are likely to churn [1]. Predicting churn is a crucial objective that helps in establishing customer retention and loyalty strategies. Along with the increasing competition in service delivery markets, the risk of customer churn is also increasing explosively. As a result, it has become imperative to implement strategies to keep track of loyal customers (non-churners). Churn models aim to identify early signals of churn and attempt to predict which customers leave voluntarily. Thus, many companies have realized that their existing database is one of their most valuable assets [2] and according to Abbasdimehr [3], churn prediction is a very useful tool to predict at-risk customers. This article is organized as follows: the next section describes the problematic. Section 3 summarizes some related work. Section 4 represents a brief review of the selected classification techniques used for this study. The different steps of our methodology are discussed in Section 4. The results are presented in section 5. And section 6 to present the results and analyze the performance of each model. Finally, section 7 concludes the article.

2- Problem Description
To overcome the above problem, the company must correctly predict the customer's behavior. Churn
management can be done in two ways: reactive and proactive.
The reactive strategy is to wait for the cancellation request received from the customer and then to offer interesting plans to retain the customer. The proactive strategy, on the other hand, prevents the customer from unsubscribing. This is because the possibility of unsubscribing is anticipated, and plans are offered to customers accordingly. This results in a binary classification problem where churners are distinguished from non-churners.
To deal with this problem, machine learning algorithms have emerged as a very powerful technique for predicting information on the previously captured database [4], including linear regression, support vector machines, naive bayes, decision trees, random forests, etc. In machine learning models, the information is then used to predict the churn. In machine learning models, after preprocessing, feature selection plays a major role in increasing the accuracy of classification. Researchers have developed a large number of approaches for feature selection that reduce dimensionality, overfitting, and computational complexity. For churn prediction, these features are taken from the given input vector and used for churn prediction. In this paper, to solve this problem, we used the following machine learning algorithms: Support vector machine, K-Nearest Neighbors, and Random Forest.
Support vector machines: This algorithm can be used in cases where there are two classes of objects (e.g., churners and non-churners). It can also be used when there are more than two classes of objects (e.g., churners, potential churners, and non-churners).
K-nearest neighbor: This algorithm is suitable for time series data, categorical data, and sparse datasets.
Random forest: This is a supervised machine learning algorithm which is based on classification and regression trees. It is designed to handle both categorical and continuous data types.

3- Related Work
This section briefly summarizes some related work proposed by leading researchers for the prediction of churn in the telecommunications sector.
The authors in [5] analyzed the variables that impact customer churn. They also conducted a comparative study of three machine learning models such as regression, neural network, and regression trees. The results showed that the decision tree performs better than the others because of its rule-based architecture. However, the accuracy obtained can be further improved by using one of the existing feature selection methods.
The authors in [6] adopted three machine learning approaches, namely support vector machine, neural network, and Bayesian networks for attrition rate prediction. A principal component analysis is taken into account in the feature selection process which allows for a reduction in data dimensions. However, the feature selection process can be improved to increase the accuracy of the classification by applying an optimization algorithm. The gain measure and the ROC curve were used to evaluate the performance.
In another study [7], the authors tried to solve the customer loss prediction problem using a support vector machine, a random forest and logistic regression. The performance of the SVM was approximately equal to that of the logistic regression and the random forest, but once optimal parameter selection was considered, the SVM outperformed the logistic regression and the random forest in terms of PCC and AUC.

Paper [8] presents an overview of all the machine learning models considered, as well as a detailed analysis of the feature selection techniques in use. The authors found that in the prediction models the decision tree had a higher efficiency than the others. In feature selection, optimization techniques also play an essential role that improves the prediction techniques. After a comparative analysis of existing methods.
In [9], the authors adopted two machine learning models, decision tree and logistic regression on a churn prediction dataset. They used the WEKA tool for experimentation. But the customer churn problem can be solved more effectively by using other machine learning methods.

4- A Brief Review of the Machine Learning Classification Algorithms Used

4-1- Bagging Support Vector Machine
Support Vector Machine or SVM is a supervised learning technique that aims to analyze data to detect patterns. There are two types of support vector machines: linear and nonlinear [10]. If the data domain can be divided linearly (e.g., straight line or hyperplane) to separate the classes in the original domain, it is referred to as a linear support vector machine. Nonlinear support vector machine is used when the data domain cannot be split linearly and can be translated to a space called the feature space where the data domain can be divided linearly to distinguish the classes [11]. On the basis of a set of training data, SVM attempts to determine the optimal separating hyperplanes between examples of distinct classes by representing observations as points in a high dimensional space. New instances are represented in the same space and assigned to a class depending on their closeness to the dividing gap [12]. Bagging, also called Bootstrap aggregating, is an ensemble learning approach that helps in the improvement of
performance and accuracy of a machine learning algorithm. It is mainly used to minimize a prediction model's variance and to deal with bias-variance tradeoffs. In [13] Various simulated results for IRIS data categorization and handwritten digit identification demonstrate that the proposed SVM ensembles with bagging significantly outperform a single SVM in terms of classification accuracy. When it comes to the customer churn issue, ensemble-learning techniques have been used as shown in [14], [15], [16].

4-2- K-Nearest Neighbors

When there is little or no prior knowledge about the distribution of the data, K-Nearest Neighbors (k-NN) classification is one of the most fundamental and straightforward classification procedures and should be one of the initial options for classification research [17]. The k-NN classification arose from the necessity to do discriminant analysis when valid parametric estimates of probability densities are unknown or impossible to calculate. The k-NN method predicts the values of new data points based on “feature similarity”, which implies that the new data point will be assigned a value depending on how closely it resembles the points in the training set. k-NN does not attempt to build an internal model, and no calculations are done until classification time. k-NN merely stores instances of the training data in the features space, and an instance’s class is chosen by the majority vote of its neighbors. The class most prevalent among its neighbors is assigned to the instance. k-NN finds neighbors based on distance utilizing Euclidian, Manhattan, or Murkowski distance measures for continuous variables and hamming distance measures for categorical data [18].

4-3- Random Forest

Random forests, also known as random choice forests, are an ensemble learning approach for classification, regression, and other tasks that work by building a large number of decision trees during training. It contains multiple decision trees, each reflecting a unique instance of the random forest's classification of data input. The random forest approach examines each case independently, selecting the one with the most votes as the chosen prediction. The classification findings in [18] suggest that Random Forest outperforms Decision Tree (J48) for the same number of characteristics and big data sets, i.e., with a higher number of instances, but Decision Tree (J48) is useful for small data sets (less number of instances). In addition to that the study in [19] shows that, the best classification model out of naive Bayes, decision tree, and random forest is random forest because of its high accuracy of 97.5 % when compared to the classification model of decision tree, which has an accuracy of 88.7 %.

5- Methodology

The steps and advantages of the proposed technique are as follows (Fig.1):

The gravitational search algorithm was used to select the features and reduce the dimensions of the data set, unlike the above existing approaches where the prediction accuracy is low due to inadequate feature selection. After data preprocessing, some of the most important machine learning techniques used for predictions, including SVM, were applied. To avoid overfitting, cross-validation was performed, unlike other techniques where the overfitting prevention mechanism is not considered. The power of ensemble learning was then utilized to optimize the algorithms and obtain better results, unlike the previously mentioned techniques where the performance of ensemble learning is not taken into account, which explains the low accuracies obtained. The algorithms were then evaluated on the test set using the confusion matrix and the AUC curve to compare the best performing algorithm for the given data set.

5-1- Presentation of the Data Set Used

The data set that we used in our experiments is “Telco Customer Churn” which is available on the Kaggle site, which contains a data set of 7043 customers. The data set includes information about:
Customers who have left the company in the last month - the column is called Churn.

The services to which each customer has subscribed: telephone, multiple lines, internet, online security, online backup, device protection, technical support and streaming TV and movies.

Customer account information: how long they have been a customer, type of contract, method of payment, electronic billing, monthly charges, and total charges.

Demographic information about customers: gender, age range, and whether they have partners and dependents.

The database consists of 21 attributes including a target value called Churn. The data set of the customer attributes alongside with their description is presented in Table 1. Table 2 presents the type of these attributes.

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CustomerID</td>
<td>Unique key</td>
<td>A code specific to each customer</td>
</tr>
<tr>
<td>Gender</td>
<td>Categorical</td>
<td>The gender of the customer</td>
</tr>
<tr>
<td>SeniorCitizen</td>
<td>Categorical</td>
<td>Whether the client is young or old</td>
</tr>
<tr>
<td>Partner</td>
<td>Categorical</td>
<td>Whether the client is married or not</td>
</tr>
<tr>
<td>Dependents</td>
<td>Categorical</td>
<td>If the client has someone dependent on him</td>
</tr>
<tr>
<td>Tenure</td>
<td>Integer</td>
<td>The number of months during which the customer is loyal</td>
</tr>
<tr>
<td>PhoneService</td>
<td>Categorical</td>
<td>Whether the customer has a telephone service or not</td>
</tr>
<tr>
<td>MultipleLines</td>
<td>Categorical</td>
<td>Whether the customer has a multitude of lines or not</td>
</tr>
<tr>
<td>InternetService</td>
<td>Categorical</td>
<td>If the customer has an internet service</td>
</tr>
<tr>
<td>OnlineSecurity</td>
<td>Categorical</td>
<td>If the customer has online security</td>
</tr>
<tr>
<td>OnlineBackup</td>
<td>Categorical</td>
<td>If the client has an online backup</td>
</tr>
<tr>
<td>DeviceProtection</td>
<td>Categorical</td>
<td>If the customer has device security</td>
</tr>
<tr>
<td>TechSupport</td>
<td>Categorical</td>
<td>If the customer has technical support</td>
</tr>
<tr>
<td>StreamingTV</td>
<td>Categorical</td>
<td>If the customer has on-demand television</td>
</tr>
<tr>
<td>StreamingMovie</td>
<td>Categorical</td>
<td>If the customer has the movies on demand</td>
</tr>
</tbody>
</table>

| Contract             | Categorical        | The contract renewal period                     |
| PaperlessBilling     | Categorical        | Whether the customer has paperless or non-paperless billing |
| PaymentMethod        | Categorical        | The payment method of the customer              |
| MonthlyCharges       | Integer            | The monthly charge of the client                |
| TotalCharges         | Integer            | The total charge of the client                  |
| Churn                | Categorical        | If the customer cancels his contract or not     |

### Table 2: Type of attributes

- Index
- Demographic Information
- Services
- Account Information
- Target

#### 5-2- Data Analysis

After importing data from the “.csv” file, the df.info() command was executed to display information about the database.

```python
<class 'pandas.core.frame.DataFrame'>
Index: 7043 entries, 7590-VHVEG to 3186-AJIEK
Data columns (total 20 columns):
    #   Column            Non-Null Count  Dtype
    ---  ------            --------------  ----
     0   gender           7043 non-null  object
     1   SeniorCitizen    7043 non-null  int64
     2   Partner          7043 non-null  object
     3   Dependents       7043 non-null  object
     4   tenure           7043 non-null  int64
     5   PhoneService     7043 non-null  object
     6   MultipleLines    7043 non-null  object
     7   InternetService  7043 non-null  object
     8   OnlineSecurity   7043 non-null  object
     9   OnlineBackup     7043 non-null  object
    10   DeviceProtection 7043 non-null  object
    11   TechSupport      7043 non-null  object
    12   StreamingTV      7043 non-null  object
    13   StreamingMovies  7043 non-null  object
    14   Contract         7043 non-null  object
    15   PaperlessBilling 7043 non-null  object
    16   PaymentMethod    7043 non-null  object
    17   MonthlyCharges   7043 non-null  float64
    18   TotalCharges     7043 non-null  object
    19   Churn            7043 non-null  object
dtypes: float64(1), int64(2), object(17)
memory usage: 1.1+ MB
```
It has been noticed that there is a problem with the types of the attributes "SeniorCitizen" and "TotalCharges" in which "SeniorCitizen" needs to be converted to a string type, and "TotalCharges" needs to be converted to an integer type. The conversion of these attributes to their appropriate types will be carried out as the first step.

```python
df['TotalCharges'] = pd.to_numeric(df['TotalCharges'], errors='coerce')
df['SeniorCitizen'].replace(to_replace=1, value='Yes', inplace=True)
df['SeniorCitizen'].replace(to_replace=0, value='No', inplace=True)
```

After the conversion, it was observed that the "TotalCharges" column has 11 missing values. It is known that the "TotalCharges" variable can be calculated by multiplying the two variables "Tenure" and "MonthlyCharges". However, for all the entries in the "TotalCharges" column, the corresponding "Tenure" value is 0, indicating that these customers are in their first month. Therefore, the value of "MonthlyCharges" will be directly assigned to them as their "TotalCharges" value.

```python
df_na = df[df['TotalCharges'].isna()].copy()
df_na['TotalCharges'] = df_na['MonthlyCharges']
df[df['TotalCharges'].isna()] = df_na
```

The database has been cleaned and is now prepared for visualization.

### 5.3- Data Visualization

#### a) Qualitative Variables

The qualitative variables were visualized using Python, as depicted in the figures below Fig. 2-Fig. 18:

Fig. 2 Male/female distribution

Fig. 3 Young/old people distribution

Fig. 4 Single/engaged people distribution

Fig. 5 Independent/ dependent people distribution

Fig. 6 Distribution of the customers having a telephone line at disposition
Fig. 7 Distribution of customers with several telephone lines available

Fig. 8 Type of the customer's internet service provider

Fig. 9 Distribution of customers with online security

Fig. 10 Distribution of customers with an online backup available

Fig. 11 Distribution of customers with a protective device

Fig. 12 Distribution of customers with technical support

Fig. 13 Distribution of customers with on-demand television available

Fig. 14 Distribution of customers with on-demand movies available
Fig. 15 Distribution of customers according to the duration of their contract

Fig. 16 Distribution of customers according to their billing

Fig. 17 Distribution of customers according to their type of payment

Fig. 18 Distribution of customers according to their loyalty

b) Quantitative Variables
The quantitative variables were visualized as shown in the figures below Fig. 19-Fig. 21:

Fig. 19 Distribution number of months of loyalty

Fig. 20 Distribution of the amount invoiced to the customer each month

Fig. 21 Distribution of the amount invoiced to the customer in total

5.4- Study of Correlation
The figures that follow represent the correlation between each of the variables and the target variable Fig. 22-Fig. 36:
Fig. 23 Age and churn correlation

Fig. 24 Marital status and churn correlation

Fig. 25 Financial status and churn correlation

Fig. 26 Telephone coverage and churn correlation

Fig. 27 Type of phone coverage and churn correlation

Fig. 28 Type of internet service and churn correlation

Fig. 29 Online security coverage and churn correlation

Fig. 30 Online backup coverage and churn correlation
The Fig. 37 shows the correlation between the churn variable and the other variables:
Based on the correlation diagram and the results obtained from the visualization of the independence of each variable with the target variable, it has been observed that there are variables with weak correlation. However, these variables will be retained as the objective is not to generalize the model, but rather to specifically fit it to this particular company. This approach may result in the model being overfitted to match the company's specific characteristics.

5-5- The Learning Phase

The learning in this study is based on three machine learning algorithms: Bagging SVM, Random Forest and the k-NN algorithm.

The first step in the learning is to convert all qualitative variables into quantitative variables.

```python
df['Churn'].replace(to_replace='Yes', value=1, inplace=True)
df['Churn'].replace(to_replace='No', value=0, inplace=True)
df_dum = pd.get_dummies(df)
```

After the execution, the "df_dum" DataFrame now consisted solely of quantitative variables.

Subsequently, the various methods and algorithms available in the "Scikit-Learn" library were implemented using Python language.

```python
from sklearn.preprocessing import StandardScaler
from sklearn.pipeline import make_pipeline
from sklearn.svm import SVC
from sklearn.ensemble import BaggingClassifier, RandomForestClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import train_test_split
```

5-6- Algorithms Implementation

Bagging SVM

```python
model = None
score = 0
for i in range(10):
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.30)
    mdl = make_pipeline(StandardScaler(),
                        BaggingClassifier(base_estimator=SVC(gamma="auto"),
                                          n_estimators=10))
    mdl.fit(X_train, y_train)
    a = mdl.score(X_test, y_test)
    if a > score:
        model = mdl
        score = a
```

K-Nearest Neighbors

```python
model = None
score = 0
for i in range(10):
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.30)
    mdl = KNeighborsClassifier(n_neighbors=10)
    mdl.fit(X_train, y_train)
    a = mdl.score(X_test, y_test)
    if a > score:
        model = mdl
        score = a
```

Random Forest

```python
model = None
score = 0
for i in range(10):
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.30)
    mdl = RandomForestClassifier(max_depth=5)
    mdl.fit(X_train, y_train)
    a = mdl.score(X_test, y_test)
    if a > score:
        model = mdl
        score = a
```

6- Results and Performance Analysis

6-1- Confusion Matrix
In order to evaluate the performance of the applied models and the prediction rate of customer churn on the test set, various metrics were utilized, including precision, recall, accuracy, and F-score. They measure the ability of the predictive models to correctly predict the customer churn. The four indicators previously mentioned are calculated from the information captured using the confusion matrix and are presented in Table 3. The representation of the confusion matrix is shown in Fig. 38. True positives and false positives are denoted $T_p$ and $F_p$, while false negatives and true negatives are denoted $F_n$ and $T_n$ [20].

- True positive ($T_p$): The number of clients who are in the churn class and whom the predictive model correctly predicted.
- True negative(523,332),(583,350) ($T_n$): The number of clients who are in the non-churners class and that the predictive model correctly predicted.
- False positive ($F_p$): The number of customers who are not churners but that the predictive algorithm identified as churners.
- False Negative ($F_n$): The number of customers who are churners but that the predictive model identified as non-churners.

A value closer to 1 implies a better combination of precision and recall achieved by the classifier.

**6-3- AUC Curve**

To evaluate the performance of the models on the positive and negative classes of the test set, we employed the AUC curve. A high value of the AUC score indicates that the model performs better on the positive and negative classes. The AUC scores achieved for the three predictive models used to predict the target variable Churn are shown in Table 3 and Fig. 39, graphically represents the AUC scores obtained for Bagging SVM, k-NN and Random Forest. According to the AUC scores, all the selected models perform well on the test set. However, the most adequate classifier is Bagging SVM with an AUC score of 84%.

![Fig. 38 Confusion matrix](image)

**6-2- Performance Indicators**

- **Recall:** The recall is the ratio of true churners or true positives ($T_p$), and is calculated as follows:
  \[
  \text{Recall} = \frac{T_p}{T_p + F_n}
  \]

- **Precision:** The precision is the ratio of predicted correct churners, its formula is as follows:
  \[
  \text{Precision} = \frac{T_p}{T_p + F_p}
  \]

- **Accuracy:** The accuracy is the ratio of the number of all correct predictions and is written as:
  \[
  \text{Accuracy} = \frac{T_p + T_n}{T_p + T_n + F_p + F_n}
  \]

- **F-score:** The F-score is the harmonic mean of precision and recall and is written as follows:
  \[
  \text{F-score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}
  \]

Table 3: Comparison of the model used

<table>
<thead>
<tr>
<th>Model</th>
<th>Accuracy %</th>
<th>Recall %</th>
<th>Precision %</th>
<th>F-score %</th>
<th>AUC Score %</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bagging SVM</td>
<td>80.26</td>
<td>80.63</td>
<td>79.71</td>
<td>80.16</td>
<td>84</td>
</tr>
<tr>
<td>KNN</td>
<td>79.03</td>
<td>76.30</td>
<td>76.30</td>
<td>76.92</td>
<td>80</td>
</tr>
<tr>
<td>Random Forest</td>
<td>79.47</td>
<td>79.44</td>
<td>78.91</td>
<td>79.14</td>
<td>82</td>
</tr>
</tbody>
</table>

![Fig. 39 Models AUC curve](image)
In the telecommunications sector, churn prediction is an issue that has attracted the interest of various researchers in recent years. It is becoming one of the sources of revenue for companies and helps to prevent customers from terminating their contracts, it opens the possibility of renegotiation in order to retain the customer by implementing retention strategies.

In this paper, we provide a comparative study of churn prediction in the telecom sector using well-known machine learning techniques such as random forest, k-nearest neighbor, and support vector machine. The experimental results show that all three machine learning techniques give high accuracy for churn prediction. The SVM and bagging classifiers as an ensemble method outperformed the other algorithms in terms of all performance measures such as accuracy, precision, F-measure, recall and AUC score. Predicting churn for a business is proving to be a very tedious task so there is stiff competition in the market to retain customers by providing services that are beneficial to both parties. In futurity, with the upcoming concepts and frameworks in the field of reinforcement learning and deep learning, machine learning proves to be one of the most used and efficient ways to overcome the problems like churn prediction with better accuracy and precision.

References


