Topics In Computer Science – Machine Learning

CSCI 6905 Spring 2018, Group 1

Naïve Bayes Classifier Algorithm

Churn Prediction for KKBOX Music Streaming Service Provider

Arjun Aneja

William Clark

Sumati Kulkarni

Babak Maleki Shoja

Venkatesh Reddy Pala

Vishwa Patel

Table of Contents

**[Learning Goals](#_30j0zll) 3**

[Churn Prediction for KKBOX Music Streaming Service Provider](#_3znysh7) 4

[Abstract](#_2et92p0) 4

[1. Business problem](#_tyjcwt) 4

[2. Descriptive and Target Features](#_3dy6vkm) 4

[3. Naive Bayes Model](#_1t3h5sf)  7

4. Algorithm 8

5. Implementation 10

[6. Results](#_4d34og8) 14

7[. Conclusion](#_2s8eyo1) 14

[**Individual Contributions**](#_17dp8vu)15

[**Team Summary**](#_26in1rg)16

[**References**](#_1ksv4uv)17

**Learning Goals**

In this assignment, the team will develop a model for KKBOX Music Streaming Service Provider as our group project to illustrate how to create and use a naive Bayes model to predict churn of a subscribed user using descriptive features provided by KKBOX. To train a naive Bayes model using this data, we need to compute the prior probabilities of the target feature taking each level in its domain, and the conditional probability of each feature taking each level in its domain conditioned for each level that the target can take. The purpose of this assignment was to get familiar with applying a naive Bayes model to an existing problem, how we can evaluate the goodness of fit and to determine whether the user will churn or not.

**Churn Prediction for KKBOX Music Streaming Service Provider**

**Abstract**

In this document, the model developed and used to predict churn of a subscriber based on descriptive features for KKBOX music streaming service provider is defined. First, we briefly review the business problem and descriptive and target features provided by the KKBOX for this problem. Then, we describe the model we used for predicting churn based on descriptive features. Next, the results of applying the naive Bayes model is provided. Finally, we draw conclusion regarding churn prediction problem.

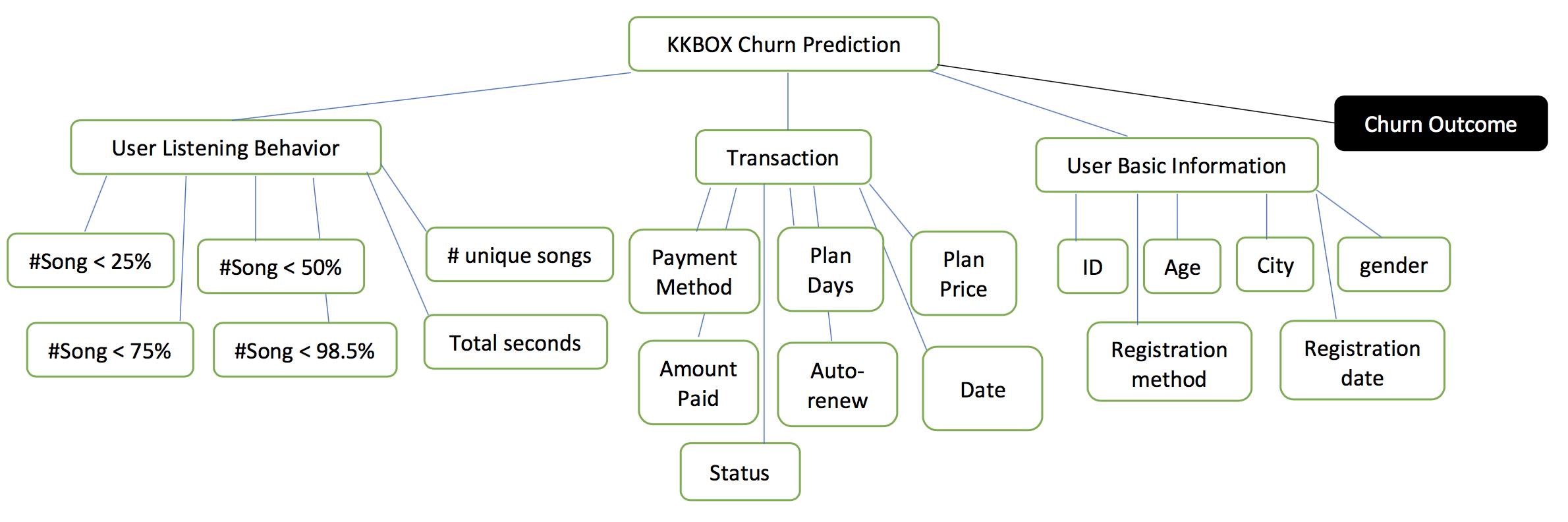
**1. Business problem**

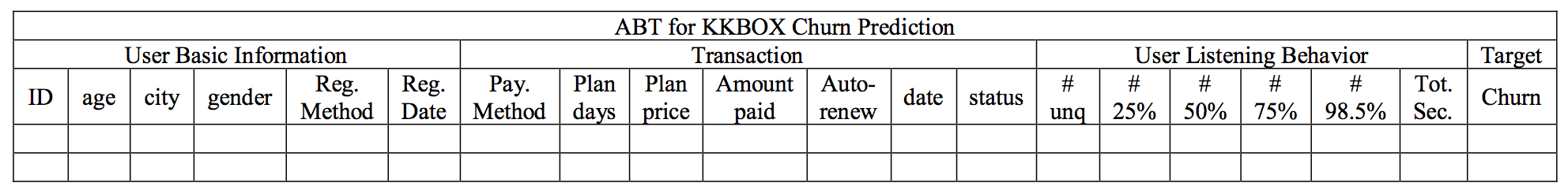
As provided in the project proposal, KKBOX, an Asian music streaming service provider, is facing the challenge of predicting whether a subscriber churn after his or her subscription expires or their decision to extend their subscription. This is a critical problem for such businesses and even a slight deviation from the predictions KKBOX has gathered a large amount of data from its users to resolve this problem.

The analytic solution to this problem defined as follows: We proposed to develop a model to predict the churn of a paid user after subscription expires. We take this into consideration to apply the prediction model and evaluate the results.

**2. Descriptive and Target Features**

Before describing the features, we should note that the prediction subject is defined as a paid subscriber. Here, we briefly review the features for the problem. The features of this problem and corresponding ABT developed as follows.





As explained in previous report, we need to predict whether a paid user churn when the subscription expires. We call this feature “Churn” which is going to be a binary feature because churn will happen (Churn value equal to 1)” or the user renew his or her subscription (Churn value equal to 0).

Descriptive features are as follows.

* IDs: the ID of a user which is unique for each user;
* Age: the age of the user which is continuous;
* City: the location of the user which is a categorical feature;
* Gender: the gender of the user which is categorical with the cardinality of 2;
* Registration method: the method user utilized to register which is categorical;
* Registration date: the date of user registration which is considered as a continuous feature;
* Payment Method ID: the method user used to pay which is categorical;
* Payment plan days: the plan the user chose which is categorical with the cardinality of the available plans;
* Plan list price: the price of each plan which is categorical with the cardinality of the available plans;
* Actual amount paid: the amount the user actually paid which is a continuous feature;
* Auto-renew: the feature denotes that if a user has activated auto renew. It is categorical feature with cardinality of 2;
* Transaction date: similar to registration date, it is continuous and determines the date payment has paid;
* Status: denotes if a user is still active or canceled subscription which is categorical with the cardinality of 2;
* Number of songs played by user less than 25% or 50% or 75% or 98.5% which is continuous;
* Number of unique songs played by the user which is considered continuous;
* Total seconds of music played by the user which is considered continuous;

**3.** **Naive Bayes Model**

**Introduction**

Naive Bayes is a simple technique for constructing classifiers: models that assign class labels to problem instances, represented as vectors of feature values, where the class labels are drawn from some finite set. It is not a single algorithm for training such classifiers, but a family of algorithms based on a common principle: all naive Bayes classifiers assume that the value of a particular feature is independent of the value of any other feature, given the class variable. For example, a fruit may be considered to be an apple if it is red, round, and about 10 cm in diameter. A naive Bayes classifier considers each of these features to contribute independently to the probability that this fruit is an apple, regardless of any possible correlations between the color, roundness, and diameter features.

For some types of probability models, naive Bayes classifiers can be trained very efficiently in a supervised learning setting. In many practical applications, parameter estimation for naive Bayes models uses the method of maximum likelihood; in other words, one can work with the naive Bayes model without accepting Bayesian probability or using any Bayesian methods.

Despite their naive design and apparently oversimplified assumptions, naive Bayes classifiers have worked quite well in many complex real-world situations. In 2004, an analysis of the Bayesian classification problem showed that there are sound theoretical reasons for the apparently implausible efficacy of naive Bayes classifiers. Still, a comprehensive comparison with other classification algorithms in 2006 showed that Bayes classification is outperformed by other approaches, such as boosted trees or random forests.

An advantage of naive Bayes is that it only requires a small number of training data to estimate the parameters necessary for classification.

**4. Algorithm**

Abstractly, naive Bayes is a conditional probability model: given a problem instance to be classified, represented by a vector representing some n features (independent variables), it assigns to this instance probabilities

for each of K possible outcomes or classes

The problem with the above formulation is that if the number of features n is large or if a feature can take on a large number of values, then basing such a model on probability tables is infeasible. We therefore reformulate the model to make it more tractable. Using Bayes' theorem, the conditional probability can be decomposed as

In plain English, using Bayesian probability terminology, the above equation can be written as

In practice, there is interest only in the numerator of that fraction because the denominator does not depend on C and the values of the features xi are given, so that the denominator is effectively constant. The numerator is equivalent to the joint probability model.

which can be rewritten as follows, using the chain rule for repeated applications of the definition of conditional probability:

Now the "naive" conditional independence assumptions come into play: assume that each feature xi is conditionally independent of every other feature xj for given the category Ck. This means that

Thus, the joint model can be expressed as

Where denotes proportionality.

This means that under the above independence assumptions, the conditional distribution over the class variable C is:

where the evidence is a scaling factor dependent only on that is a constant if the values of the feature variables are known.

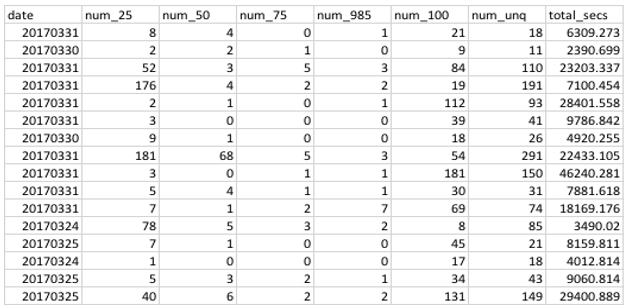
The discussion so far has derived the independent feature model, that is, the naive Bayes probability model. The naive Bayes classifier combines this model with a decision rule. One common rule is to pick the hypothesis that is most probable; this is known as the maximum a posteriori or MAP decision rule. The corresponding classifier, a Bayes classifier, is the function that assigns a class label.   for some k as follows:

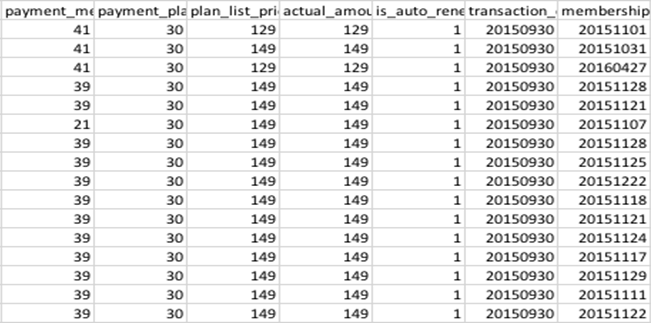
**5. Implementation**

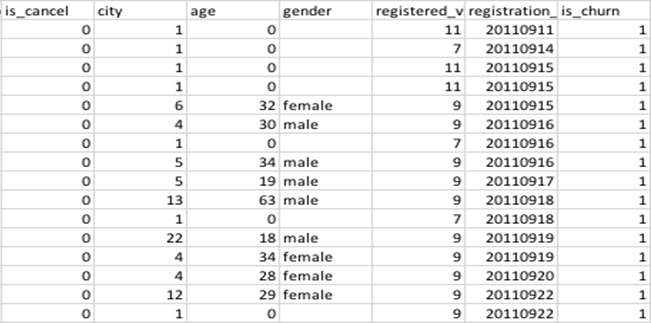
We implemented above described algorithm on our dataset. We used Python and used following code to train and evaluate the model on our dataset.

|  |
| --- |
| # -\*- coding: utf-8 -\*-  """  Created on Sat Feb 24 18:33:39 2018  @author: patel  """  import pandas as pd  import numpy as np  import matplotlib.pyplot as pl  from sklearn import preprocessing,cross\_validation,svm  from sklearn.cross\_validation import train\_test\_split  from sklearn.metrics import confusion\_matrix  from sklearn.externals.six import StringIO  from sklearn import tree  from matplotlib import style  from matplotlib.colors import ListedColormap  style.use("ggplot")  dataset = pd.read\_csv('test.csv')  dataset = dataset.replace(['male','female'],  [0,1])  #print(dataset)  # Seperating value into two objects  #X = dataset.iloc[:, [2,3,4]].values  #A = dataset.iloc[:, 17].values  X = dataset.iloc[:, :-1].values  y = dataset.iloc[:, 21].values  #dataset[X[:, 17]] = dataset[X[:, 17]].replace(0, dataset[X[:, 17]].mean())  #dataset[A] = dataset[A].replace(0, dataset[A].mean)  print("This is new age column replace with male and female into 0 and 1.")  #print(A)  print(X)  print(y)  print("This is the new value of gender column with missing values.")  print(X[:, 18])  # Replacing nan value with median in gender column which is 18th column  from sklearn.preprocessing import Imputer  imputer = Imputer(missing\_values='NaN', strategy = 'median', axis = 0)  #X[:, [18,19]] = X[:,[18,19]].reshape(1, -1)  imputer = imputer.fit(X[:, [18,19]])  X[:, [18,19]] = imputer.transform(X[:, [18,19]])  print("This is gender column after replacing missing values with median function")  print(X[:, 18])  print("\n")  # Split the data into training and testing part  from sklearn.cross\_validation import train\_test\_split  # Giving 20% of total dataset in testing part and remaining in training part (80%)  x\_train, x\_test, y\_train, y\_test = train\_test\_split(X,y,test\_size=0.20, random\_state = 0)  # Implementing Naive-Bayes Classifier algorithm for our dataset  from sklearn.naive\_bayes import GaussianNB  #Create a Gaussian Classifier  model = GaussianNB()  # Train the model using the training sets  model.fit(x\_train,y\_train)  #Predict Output  predicted= model.predict(x\_test)  print("This is the predicted data for testing X which contains 2000 datasets")  print(predicted)  print("\n")  accuracy = model.score(x\_test, y\_test)  print("This is the accuracy of Naive Bayes Classifier algoritham")  print(accuracy)  print("\n")  # Calculating confusion matrix for machine learning prediction and  cm= confusion\_matrix(y\_test, predicted)  print("The confusion matrix is described below.")  print(cm) |

We used total data to split into training and test. Some instances of training data are shown below:







We used our data set and the model is evaluated by a test dataset with 2000 instances. The results statistics are as follows.

Using this data and using python’s inbuilt modeling tool we can build prediction model that would assign a conditional probability to each level of every feature and final prediction is based on the formula shown below:

**6. Results**

**Table 1 Results of Bayes Classifier Algorithm**

|  |  |
| --- | --- |
| **Results Statistics** | |
| Number of instances | 2000 |
| Number of true predictions | 1745 |
| Number of false predictions | 254 |
| False prediction percentage | 12.7 |
| **Model accuracy (%)** | **87.3** |

As demonstrated in the results and shown in Table 1, the model shows a good performance in predicting the Churn using available data. However, comparing to K-Neighbors Classifier we used before, K-Neighbors Classifier outperforms Bayes Classifier Algorithm. Consequently, the K-Neighbor Algorithm Classifier is a more suitable algorithm for our problem and available dataset.

**7. Conclusion**

The Naïve Bayes Model shows a good performance in predicting when the user will Churn. As it is not the best when compared to K-Neighbors Classifier, and also as K-Neighbors is more suitable algorithm for our type of dataset, we proceed with using K-Neighbors algorithm.

**Individual Contributions**

|  |  |
| --- | --- |
| **Arjun Aneja** | Provided invaluable contributions to the completion of the tasks assigned to the group  Reviewed Report |

|  |  |
| --- | --- |
| **William Clark** | Provided invaluable contributions to the completion of the tasks assigned to the group  Reviewed Report |

|  |  |
| --- | --- |
| **Sumati Kulkarni** | Provided invaluable contributions to the completion of the tasks assigned to the group  Drafted Report  Reviewed Report |

|  |  |
| --- | --- |
| **Babak Maleki Shoja** | Provided invaluable contributions to the completion of the tasks assigned to the group  Drafted Report  Reviewed Report |

|  |  |
| --- | --- |
| **Venkatesh Reedy Pala** | Provided invaluable contributions to the completion of the tasks assigned to the group  Drafted Report  Reviewed Report |
| **Vishwa Patel** | Provided invaluable contributions to the completion of the tasks assigned to the group for this project  Drafted report  Reviewed Report |

**Team Summary**

This phase of the project gave us the insight in implementing predictive models and evaluate them. Moreover, we understood we need to exclude some of the features to get the results. The importance of the evaluation for the model was investigated and it was a great experience to see how machine learning can solve real-world problems and challenges.

References

* Kelleher, John, Namee, Brian Mac, Arcy, Aoife D’. (2015). Fundamentals of Machine Learning for Predictive Data Analytics*.* The MIT Press Cambridge, Massachusetts London, England
* Naive Bayes classifier. Wikipedia, the free encyclopedia <https://en.wikipedia.org/wiki/Naive_Bayes_classifier>