Topics In Computer Science – Machine Learning

CSCI 6905 Spring 2018, Group 1

Prediction Model - 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) 5

[3. Prediction Model](#_1t3h5sf) 7

[4. Results](#_4d34og8) 8

[5. Conclusion](#_2s8eyo1) 13

[**Individual Contributions**](#_17dp8vu) **14**

[**Team Summary**](#_26in1rg) **15**

[**Implementation of Code is done as follows in the Jupyter Notebook**](#_w2yv4e3covez) **16**

[**References**](#_1ksv4uv) **20**

# **Learning Goals**

In this assignment, the team developed a model for KKBOX Music Streaming Service Provider as our group project problem in order to predict churn of a subscribed user using descriptive features provided by KKBOX. The purpose of this assignment was to get familiar with applying a prediction model to an existing problem, how we can evaluate the goodness of fit and to determine features that can be excluded from the data.

## **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 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.

## 

## 

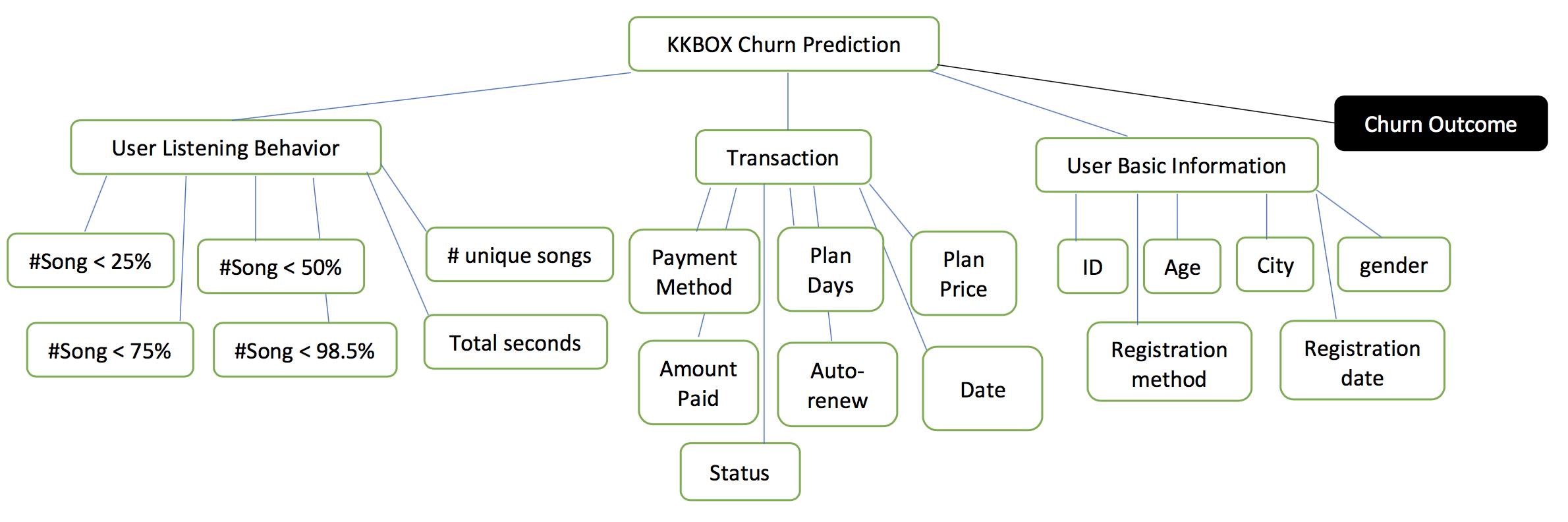
## 

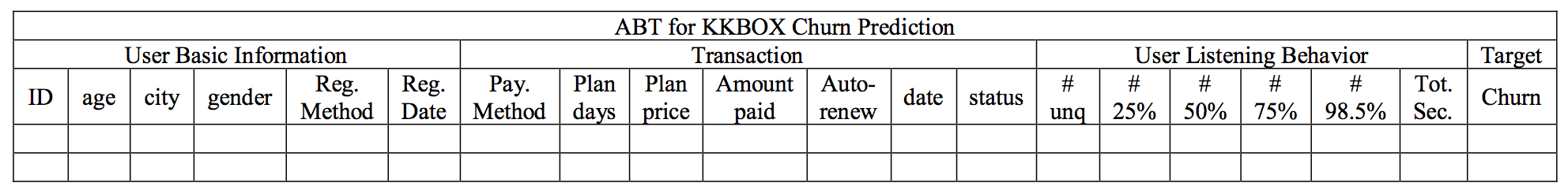
## 

## 

## **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 Feature | Description | Type |
| ID | The ID of a user which is unique for each user | Int, continuous |
| AGE | Age of user | Int, continuous |
| CITY | Location of User | Categorical |
| Gender | Gender of User | Categorical with Cardinality of 2 |
| Registration method | Method user used to register for this service | categorical |
| Registration Date | Date on which User registered | Continuous |
| Payment Method ID | Method used by user to pay | Categorical |
| Payment plan days | Plan chosen by user | Categorical with relation to available plans |
| Plan List Price | Price of plan | Categorical with relation to available plans |
| Actual amount paid | Amount user actually paid | Continuous |
| Auto-renew | The feature denotes that if a user has activated auto renew | Categorical feature with cardinality of 2 |
| Transaction date | Similar to registration date | Continuous |
| Status | Denotes if a user is still active or canceled subscription | Categorical with the cardinality of 2 |
| Number of songs | Songs played by user less than 25% or 50% or 75% or 98.5% | Continuous |
| Number of unique songs | Unique songs played by the user | Continuous |
| Total seconds of music | Total seconds of music played by the user | Continuous |

### **3. Prediction Model**

For our dataset (KKBOX customer churn prediction), we have used python language and Spyder IDE for obtaining results. Prediction model steps that we have done to achieve required results is described below:

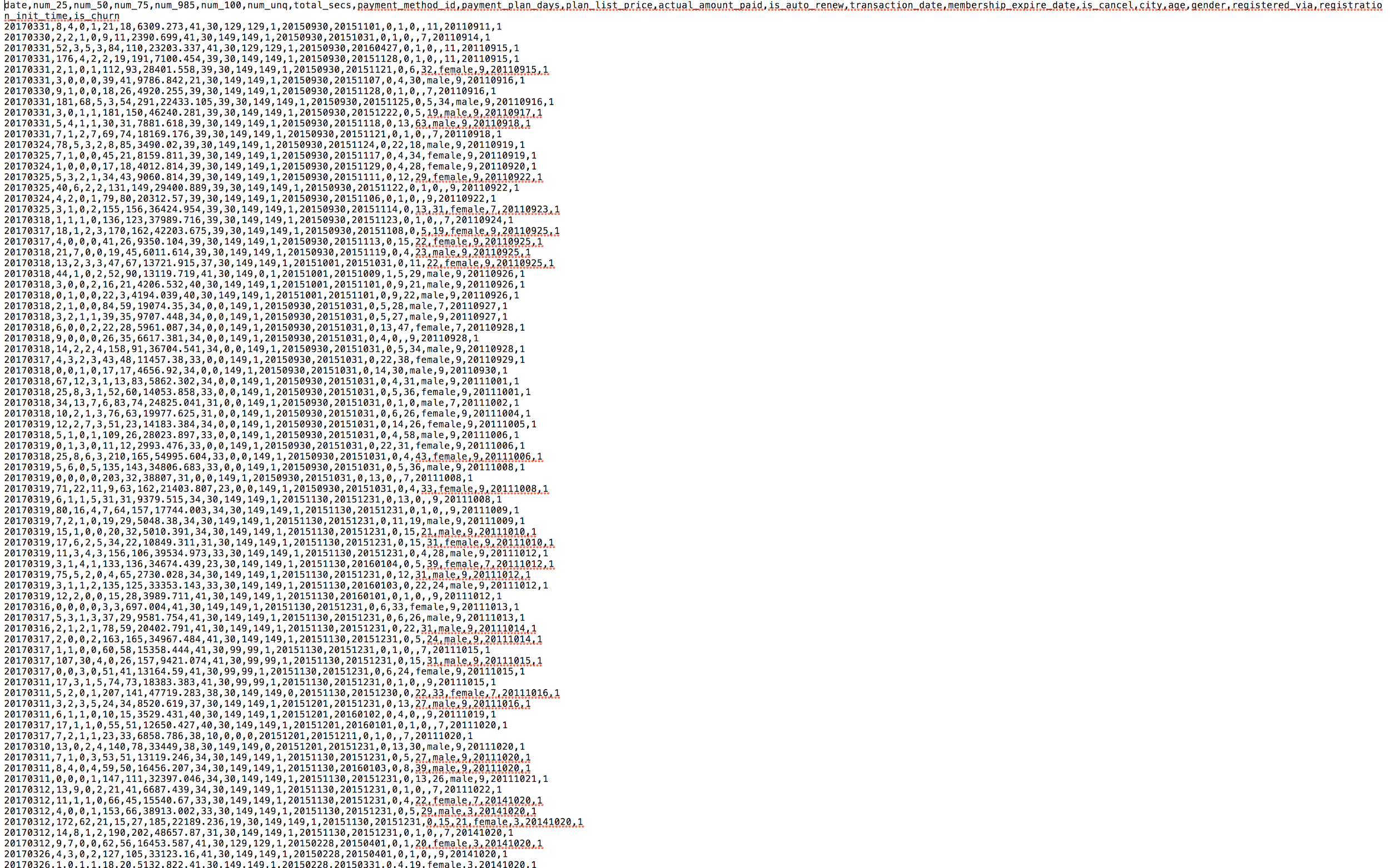
1. Firstly, we copy (read) the data from .csv file format and store it in a variable.
2. We create two different variables, X and y in which descriptive features and target feature is stored respectively
3. We used “sklearn” library, preprocessing package and LabelEncoder class which provides functionality to convert Categorical data into Numerical data (from string to int). There were some missing values in the column “source\_screen\_name”, those values are calculated to numerical value, as in our case ‘nan’ is converted to numerical value “17”.
4. Now the derived data is then converted into normalized data and then stored into same variable.
5. The normalized data is stored in an external file “.csv”.
6. Using train\_test\_split class in sklearn.cross\_validation package, we have split dataset into training sets and testing sets. In which we have given 20% of the total dataset is given into the testing part while the rest of the data is given into the training part.
7. We have used K\_neighboursclassifiers algorithm to train the machine, which includes both descriptive features and target features.
8. Once the training is done, then we have given the x\_test (descriptive features) to predict the outcome based on what model (machine) has learned in training.
9. We have then computed the accuracy of the data by comparing the actual target feature and what we received by our machine learning model.
10. To see how many features are predicted right and how many predicted wrong, we have used confusion matrix to achieve that.

### 

### **4. Results**

Dataset we have used in our model is 10,000 data. In which 8000 is given in training part and 2000 is given in testing phase. Following screenshots are our original dataset and the obtained result:

1. Original Dataset of our project:



The original test data that we have used this project, looks like the following which is provided by KKBOX. From the above data which is normalized into binary data we will predict the Churn of a customer form the above data.

1. To get Started, We will import necessary packages and libraries that will be necessary to make the predictive analytics

“

*# Dependencies of the Project*

*import pandas as pd*

*import numpy as np*

*import matplotlib.pyplot as pl*

*# Various tools that are used to process the K-NN*

*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*

”

We have used scikit-learn and tools to develop this model. In the second step we got a error which is taken care of it is because of the dependencies but the package is necessary which is automatically updated.

2. Loading the dataset and preprocessing it

“

*# Dataset that has to be imported*

*dataset = pd.read\_csv('test.csv')*

*# Male and Female into Binary data*

*dataset = dataset.replace(['male','female'],*

*[0,1])*

”

First we read the dataset which is in the same directory and then process the category data is converted to binary.

Then, we separated the features from the labels

“

*# Separating value into two objects*

*#X = dataset.iloc[:, [2,3,4]].values*

*X = dataset.iloc[:, :-1].values*

*y = dataset.iloc[:, 21].values*

”

3. Now let us see some feature values,

“

*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(X[:, 18])*

”

The result we see is as follows,

[1.93352923e-06 1.93352923e-06 1.93352923e-06 ... 1.98310690e-06  
 1.98310690e-06 1.98310690e-06]

Now time to normalize the data and then print the values of X

“

*# Normalizing the data from (0 to 1)*

*X = ((X-X.min())/(X.max()-X.min()))*

*print(X)*

”

We see the following

[[1.00000000e+00 2.33015061e-06 2.13183992e-06 ... 1.93352923e-06  
 2.47888363e-06 9.97054095e-01]  
 [9.99999950e-01 2.03268458e-06 2.03268458e-06 ... 1.93352923e-06  
 2.28057294e-06 9.97054243e-01]  
 [1.00000000e+00 4.51156821e-06 2.08226225e-06 ... 1.93352923e-06  
 2.47888363e-06 9.97054293e-01]  
 ...  
 [9.99999306e-01 2.47888363e-06 2.08226225e-06 ... 1.98310690e-06  
 2.08226225e-06 9.99032541e-01]  
 [9.99999306e-01 2.42930596e-06 1.93352923e-06 ... 1.98310690e-06  
 2.08226225e-06 9.99032541e-01]  
 [9.99998959e-01 1.93352923e-06 1.93352923e-06 ... 1.98310690e-06  
 2.08226225e-06 9.99032541e-01]]

4. Preparing the training and testing set

“

*# 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)*

”

5. Applying K-NN and defined distance metrics, which is minkowski where p=2. It is also known as Euclidean Distance equation.

“

*from sklearn.neighbors import KNeighborsClassifier*

*classifier = KNeighborsClassifier(n\_neighbors = 5, metric = 'minkowski', p = 2)*

*classifier.fit(x\_train, y\_train)*

”

We get the output as follows

KNeighborsClassifier(algorithm='auto', leaf\_size=30, metric='minkowski',  
 metric\_params=None, n\_jobs=1, n\_neighbors=5, p=2,  
 weights='uniform')

6. Implementation of KNN

“*# Giving X\_test in predicting the result of y\_pred*

*y\_pred = classifier.predict(x\_test)*

*print(y\_pred[:])*

*np.savetxt("y\_prediction.csv",y\_pred,delimiter=',')*

”

7. Calculating the Accuracy

“

*# Calculating the accuracy of the algorithm by comparing the actual target feature and what we received by our machine learning model*

*accuracy = classifier.score(x\_test, y\_test)*

*print("This is the accuracy of our algorithm")*

*print(accuracy)*

”

This is the accuracy of our algorithm  
0.934

8. Validating the result by Confusion Matrix

“*cm= confusion\_matrix(y\_test, y\_pred)*

*print("The confusion matrix is described below.")*

*print(cm)*

”

We get the result as follows

The confusion matrix is described below.  
[[1815 46]  
 [ 86 53]]

9. Testing

“

*X\_test2 = np.array([0.999999,2.08226e-06,1.93353e-06, 1.93353e-06, 1.93353e-06, 2.13184e-06, 2.231e-06, 4.77028e-05, 3.7679e-06, 3.42086e-06, 9.3206e-06, 9.3206e-06, 1.98311e-06, 0.999032, 0.999037, 1.93353e-06, 1.98311e-06, 1.93353e-06, 1.93353e-06, 2.13184e-06, 0.999538*

*])*

*X\_test2 = X\_test2.reshape(1, -1)*

*y\_pred2 = classifier.predict(X\_test2)*

*print("We inserted query into model and it predicted following decision")*

*print(y\_pred2)*

”

We inserted query into model and it predicted following decision  
[0]

Therefore we can say that from the given query the customer will not churn.

Looking at matrix, we can say that the matrix (diagonally) the sum of (106+1479) = 1585, this is the accuracy of our model, which means that out of 2000 dataset which we have given in the testing phase, 1585 data was predicted correctly. And remaining dataset which is (69+346) = 415 dataset were predicted wrong by our model (which is the remaining i.e 1.0000 - 0.7925 = 0.2175%)

### 

### **5. Conclusion**

In this assignment, the team reported how we implemented prediction model for solving the predictive analytics solution defined for the real problem of KKBOX music stream service provider. We first reviewed the problem and explained the analytics solution for the real-world problem. Then, we reviewed all the features including descriptive and target features. We determined that which feature is considered as a categorical feature and which one is considered as continuous based on their nature. The prediction model is described and trained by data. In addition, the error of the model in prediction churn is provided. The accuracy of the model is close to 80% which is a good accuracy based on the fact that we used a portion of the data for the model. It can be expected that using the whole data increases the accuracy of the model. KKBOX can use this model to predict the churn of the subscribed users.

# 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  Reviewed Report |

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

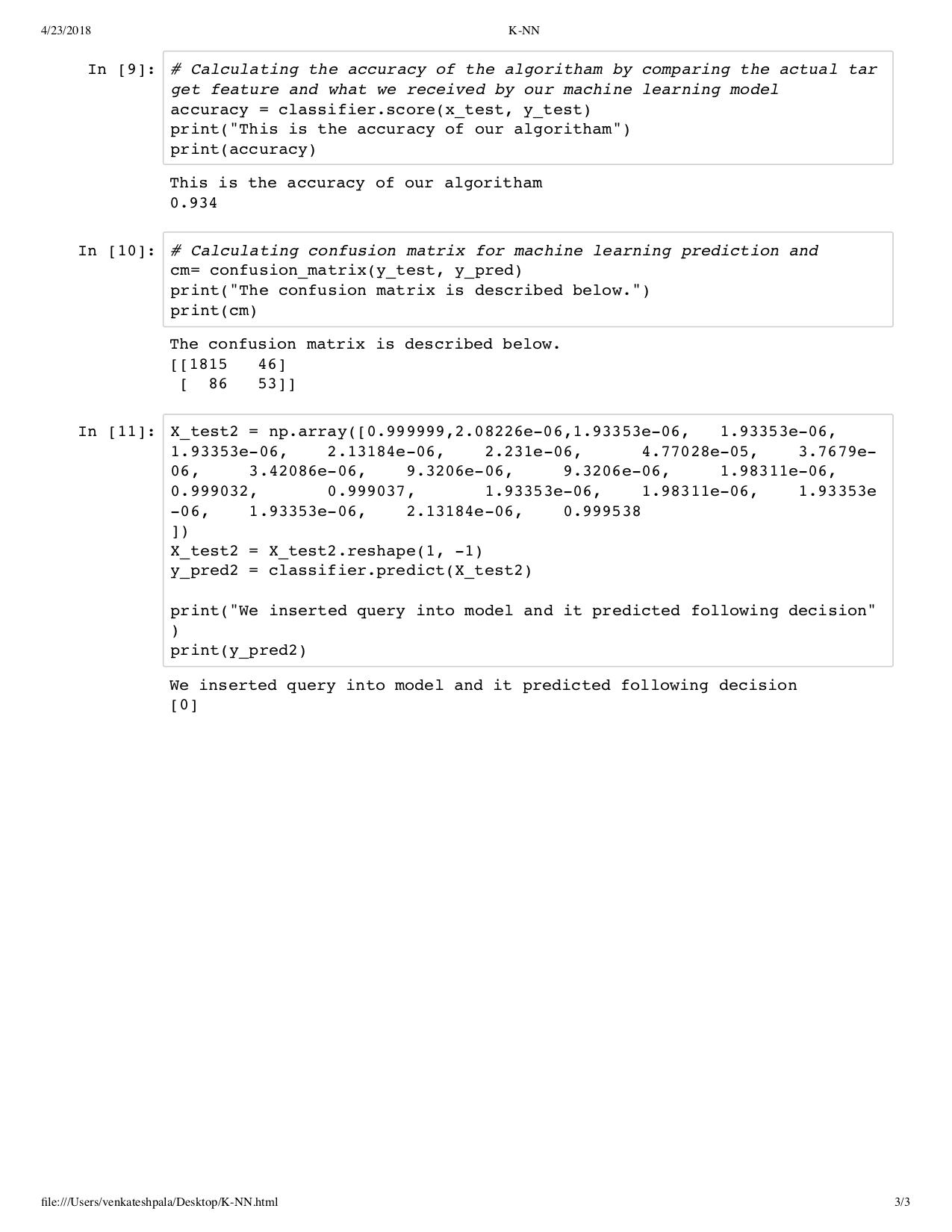
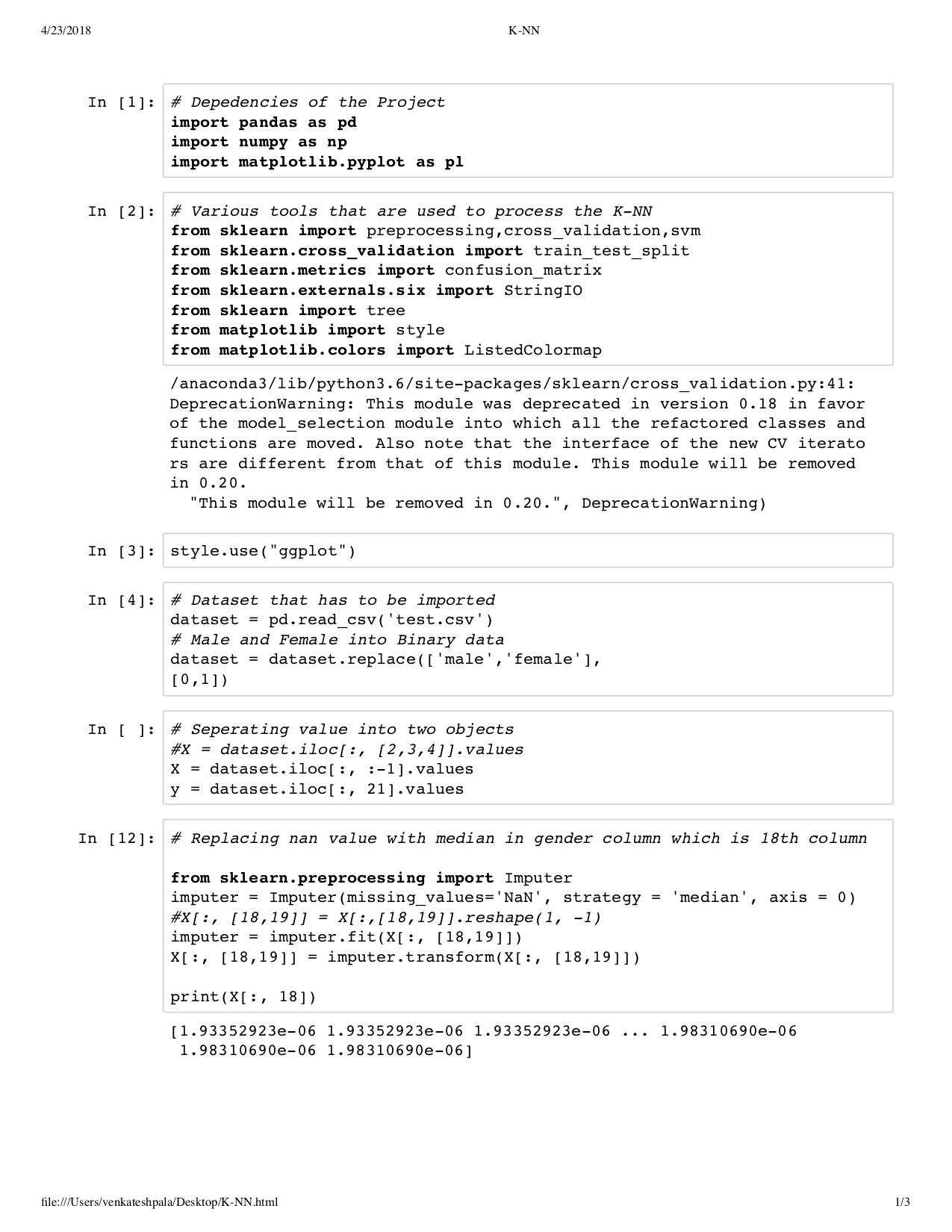
# Team Summary

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.

# 

# 

# Implementation of Code is done as follows in the Jupyter Notebook



# 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
* Kaggle, WSGM- KKBox’s Churn Prediction Challenge <https://www.kaggle.com/c/kkbox-churn-prediction-challenge#description> 2018 Kaggle Inc