WSDM - KKBox's Churn Prediction Challenge

Can you predict when subscribers will churn?

CS 6375- Machine Learning Project Report

Asok, Anagha AXA151631
Kolanpaka, Ravikiran RXK171530
Ramaswamy, Swetha SXR178830
Reddy, Sathya Pooja Rami SXR176830

December 2017

Contents

List of Figures

1. INTRODUCTION

For a subscription business, predicting the churn accurately is important because slight variations in churn can affect the profit drastically. KKBOX is a leading music streaming service who offers their service supported by paid subscriptions and advertisements. Accurate prediction of churning is important for similar establishments in ascertaining their long-term success. The existing algorithm is based on survival analysis techniques to determine the residual membership life time for each subscriber. This project aims at finding out an algorithm which can determine more accurate reasons why a user leaves and thus helps KKBOX to be proactive and maintain the subscribers.

It is planned to submit the project in Kaggle for "WSDM - KKBox's Churn Prediction Challenge -Can you predict when subscribers will churn?" It is a research competition.

Each user's behavior on the website is available and it is expected that analysis of the behavior can be lead to a conclusion stating which users would leave and who would stay. Initial analysis of data was interesting, and it is highly possible that there exists a correlation between some of the features and decision of a user to churn or continue. It is planned to attempt different machine learning algorithms and techniques like ensemble methods and Neural Networks to execute the challenge. The data is analyzable, and the idea is not so quixotic. Hence this challenge was chosen. It is a classification problem which lets the learner test the models that has been discussed in the class and evaluate the results using various scoring parameter and metric evaluations. The project aims at building an algorithm to predict whether a subscription user will churn using a dataset from KKBOX.

2. BACKGROUND

2.1 Scikit- Learn

Scikit Learn is a machine learning library which is written in Python. It has simple and efficient tools for performing a varied range of operations pertaining to data mining and data analysis. It can be considered as a repository for various classification, regression and clustering algorithms including support vector machines, random forests and many more. It was designed to operate in correspondence with Python's other numerical and scientific libraries specifically Numpy and Scipy. This library is focused on the modelling part of the data and not on the loading, manipulation and summarization of it.

2.2 Pandas

Pandas is an open source tool which can be used in Python code for data analysis. Pandas allows to carry out data analysis workflow in Python . Pandas name is derived from panel data which basically means multidimensional structured data sets. Some of the library's features are as follows: DataFrame is used for data manipulation with integrated indexing. It can also be used for dataset merging and joining. Data structure column insertion and deletion can also be performed.

2.3 Numpy

It is a package used for scientific computing with Python. Numpy can also be used as an efficient multi-dimensional container of generic data. It allows to integrate with many databases. It is an open source tool.

2.4 ROC Curve

ROC is a curve which is plotted as true positive rate(tpr) against the false positive rate(fpr) considering many threshold values. The true-positive rate is also known as the probability of detection in machine learning. The ROC analysis provides tools to select the most optimal solutions in the form of models and remove the suboptimal solutions. This curve is also known as relative operating characteristic curve specifically because it is a comparison of two characteristics which are the operating factors. Sklearn has the roc_curve function which allows for the easy implementation of the same.

2.5 Precision and Recall

The recall is basically a measure which can predict the correct instances out of the total relevant instances. Whereas precision is the fraction of relevant instances among the retrieved instances. It can be said that precision and recall are totally dependent on the factor of relevance. An instance being, if more relevant results were given by an algorithm, in that case it can be said that the algorithm has high precision. Whereas high recall can be interpreted as getting most of the relevant results.

2.5 Random Forest Algorithm

Random Forest Classifier is a powerful machine learning algorithm. It is a type of ensemble technique applied to decision trees. This algorithm can be used for both classification and regression problems. Random forest classifier randomly selects different subsets from the training data to build the decision trees and more over the features selected at each split are also selected randomly. This model uses bootstrapping to create the n number of subsets.

The basic parameters considered while creating a random forest model are the total number of trees to be generated(n), the number of features in each subset, minimum split, split criteria etc. The Algorithm has two phases when creating a model. The training phase and the testing phase.

In the training phase n number of subsets are randomly selected using bootstrap method and then the decision trees are built for each of the data and the number of attributes at each split are also selected in random.

In the test phase the prediction is made for the test data by taking the majority for the classification problems and by calculating the mean for regression problem

The advantages of the Random Forest algorithm are it reduces the problem of overfitting. This algorithm helps in both classification task and regression task. This algorithm can also be used for feature engineering, extracting or identifying new features from the given data.

One negative side to this model is it is difficult to interpret making it difficult for people to

understand why a prediction has been made. It is also not suitable for all types of classifications like text classification involving very high dimensional features.

2.6 Class Imbalance

Class Imbalance problem in machine learning can be explained as the problem that arises when the total number of instances of one class of data is far more or far less than the total number of instances of the other class. In many cases when provided with such kind of data it leads to false predictions and false accuracy. Many conventional algorithms tend to lean towards the majority class irrespective of the data distribution. The different ways to tackle the imbalanced data is cost function based approach or sampling based approach. The idea behind the cost based functions is if we think one false negative is worse than false positive, we will consider the false negative as hundred false negatives.

The sampling based approach has Oversampling, Under sampling and combination of the both. Oversampling randomly replicates the minority class to increase the number of the minority instances and balance the data. Under sampling means it removes some of the instances of the majority class to reduce its effect on the algorithm.

Another method called synthetic minority over sampling (SMOTE) is widely used to tackle with class imbalance problem. It is simple and effective. It is a combination of both over sampling and under sampling, but here the over sampling is not done by replicating the minority class. Instead a new minority class is developed using appropriate algorithm.

3. PROBLEM DEFINITION AND ALGORITHM

3.1 Data Description

The challenge in hand is the churn prediction challenge by WSDM. The task has a varied set of aspects that are to be taken into consideration. The outcome is to predict whether a user would churn when their subscription is over at the end of the month. The user's tendency to re-establish their subscription can be spread across multiple factors and features as follows. The different datasets are the inputs which must be considered for the prediction of the outcome.

Since it's a music service provider, they have a list of members. They have a monthly subscription plan. After 30 days, these members can choose to renew or cancel the subscription. There is also a possibility of auto- renewal which can also be cancelled. As the subscription model of the company is diverse, they have identified transaction date, membership expiration date, and is_cancel as the key features. The data is provided as five different files and their complete description and of their features are as follows.

Train.csv data is selected from the users whose membership would expire in one month's time. The month of February is taken into consideration. The churn renewal is basically

predicted for the month of March of the year 2017.The train data set has the following labels: user id and is_churn.

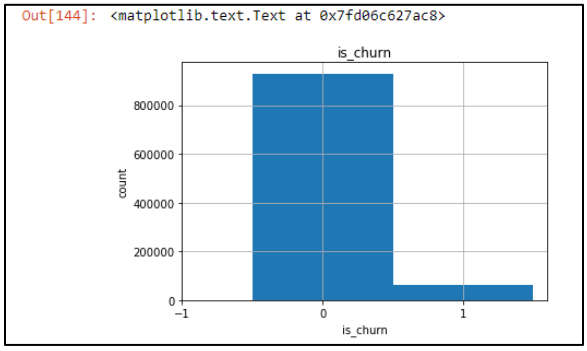


Fig 1: Users vs Churn

Only around 6000 people have churned so far. The aim is find out when will remaining people churn.

Transactions.csv dataset has the information regarding all the transactions the user performed until February of 2017. The columns in this dataset are msno, payment_method_id, payment_plan_days, plan_list_price, actual_ amount_paid, is_auto_renew, transaction_date ,membership_expire_date and is_cancel.

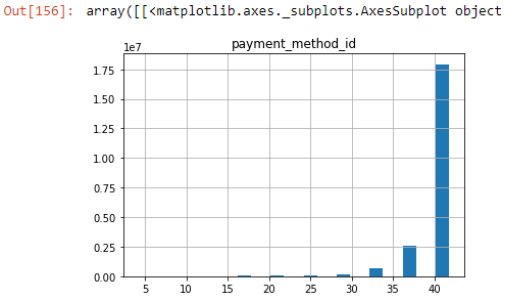


Fig 2: Users vs Payment_ID

Payment ID 41 is the most frequently used ID number. In the payment_plan_days column, the value is 450 days if the user has opted for either manual or automatic re-subscription. The column named as auto-renew has binary values.

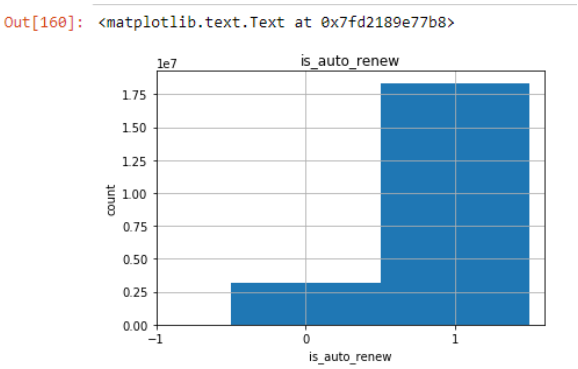


Fig 3: Count of users who auto-renewed

Majority of users renew their service automatically. Feature is_cancel is also binary which is used to determine whether this particular user has cancelled or not.

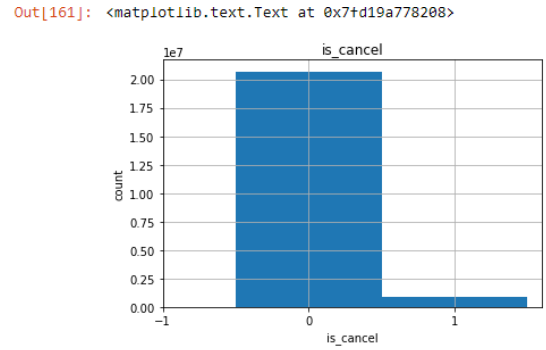


Fig 4: Count of users who cancelled

It can be noticed that majority of users have not cancelled the subscription. For deciding whether a user has churned or not can be very well dependent on the transaction_date and membership_explire_date features whose difference can give the period of time a user has been on the website.

Member.csv dataset has the information regarding the users of the subscription service. The features are msno, city, bd, gender, registered_via. registration_init_time, and expiration_date. The data under the feature city is encoded wherein 21 cities are included.

```
Out[149]: <matplotlib.text.Text at 0x7fd1efeed8d0>
```

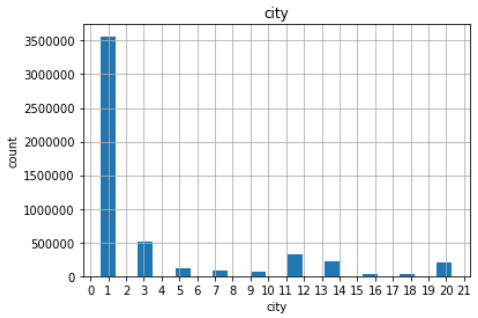


Fig 5: Cities and Users

The feature bd has information regarding the age of a particular user. Gender feature has catogorical data and a lot of information in this section is missing.

```
Out[148]: <matplotlib.text.Text at 0x7fd0ed1d9a90>
```

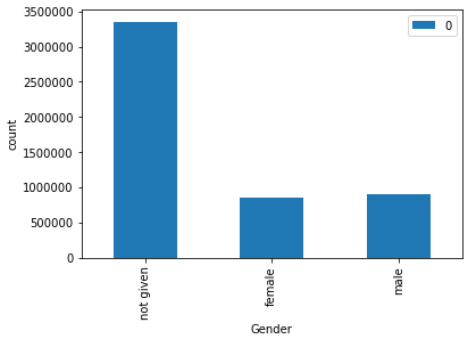


Fig 6: Gender of the users

The feature registered_via is also in an encoded format.

```
Out[152]: <matplotlib.text.Text at 0x7fd19810b390>
```

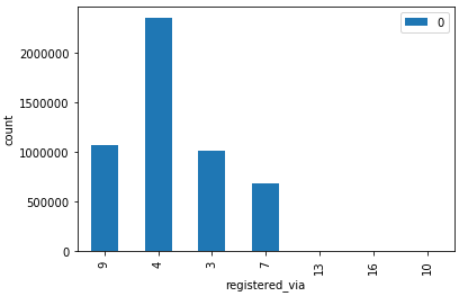


Fig 7: Registered via Vs No: users

The registration_init_time is a date encoded as an integer. In order to use this particular information, it needs to be decoded to date format.

User_logs.csv is another dataset with information regarding the user's behavioral characteristics. The features are msno, date, num_25, num_50, num_75, num_985, num_100, num_unq and total_secs. This dataset has information from the month of Jan 2015 to Feb 2017. The features num_unq has the number of unique songs the user played on that day. The feature total_secs has the total seconds the user played songs on that particular day. The feature num_25 has the number of songs played for less than 25% of the time frame. The feature num_50 has the total number of songs that were played between 25% and 50% of the song's length.

Test.csv is same as that of train data set. The two columns are msno and is_churn_column.

3.2 Algorithm Definition

The dataset for the churn prediction challenge is diverse and has many aspects that must be considered. The algorithm that we have developed takes a holistic view in the diverseness of the data set. The data is huge, and it contains millions of records. We have come up with some techniques to reduce the file size before pre-processing it.

Also, we have decided to engineer new features from the existing ones which can be more meaningful like discount from list price and paid amount etc. After feature engineering the different datasets have to be combined on the basis of ID of the users and it has to be again

combined with test and train data. Correlation of features need to be analyzed to identify best features. It must be then executed using various models like Random Forest and Artificial Neural Network. The results should be evaluated based on precision, recall, ROC curve etc. to find out the best model.

4. Experimental evaluation

4.1 Methodology

4.1.2 Data Pre-Processing

The dataset for the churn prediction challenge is diverse and has many aspects that must be considered. The algorithm that we have developed takes a holistic view in the diverseness of the data set. The data is huge, and it contains millions of records. We have come up with some techniques to reduce the file size before pre-processing it. We have checked all the datasets for missing and null values. All the ones that had any incomplete values were discarded.

4.1.3 Memory Reduction

The datatype of the features has been changed to reduce memory. The values have a huge range and thus, considering the upper bound, the range of the entire column had been set to the upper bound of the range which increased the memory size of the datasets. After calculating the minimum and maximum range of the columns, if the value lies between -128 and 127 then in that case the datatype is converted to int8. If it is in the range from 32767 to -32768, then in case it gets converted to int16. If it is in the range from 2147483647 to -2147483648, then in that case it gets converted into int32. The datatypes of 'Members' Data frame have been loaded as int8 and int 16 to reduce the memory. This dataset contains the information of the users, such as city,bd(age), gender etc.

This gave reduction of size of memory of the files.

```
In [10]: get_memory_usage_datafame()
Out[10]:
```

	DataFrame	Memory MB	Records
0	members	309.881958	6769473
1	train	29.966675	1963891
2	test	13.846985	907471
3	transactions	1577.825966	22978755

Fig 8: Data frames Before Memory Reduction

```
In [8]: get_memory_usage_datafame()
Out[8]:
```

	DataFrame	Memory MB	Records
0	members	180.764507	6769473
1	train	16.856288	1963891
2	test	13.846985	907471
3	transactions	723.170276	22978755

Fig 9: Data frames After Memory Reduction

4.1.4 Pre-Processing of Members

4.1.4.1 Birth Date / Age

The feature 'bd' which indicates the age of the users have numerous outliers present. Some of these values in this data are negative. Also, some of the values for this column are greater than 1000. Hence, the values for this data have been clipped(thresholding). A lower threshold of 0 has been put and a higher threshold of 100 has been

put. Also, after clipping the data has been categorized. Numerical variables are converted into categorical using binning. A Label Encoder was used for it and it has been observed that vast number of users in the data set are teenagers. The binning is performed as:

Age group of 0-18 = Children, 18-30 = teenagers, 30-50 = medium, 50-100 = old

4.1.4.2 Gender

Many users have not provided their gender. The column gender has been converted into a numerical variable, with male = 1 and female = 2. All the null values in this case have been replaced with 0. It is planned to present it as an additional categorical value.

4.1.4.3 Feature Engineering

It has been observed that around 14000 users from the test data are not present in the members data. Therefore, Preprocessing needs to be done for those users.

The features are checked for null values. The feature registered_via has negative value in one of the rows and it has been remove. Feature City is a categorical attribute and hence all the values in this column have been replaced with mode of that attribute. After converting feature bd value into categorical value, the null values have been replaced with mode of that attribute. The feature bd in the case of numerical attribute has been replaced with mean of the whole column.

4.1.5 Pre-Processing and feature Engineering of Transactions

4.1.5.1 New Features

Th data frame Transactions had multiple features which on analysis was found out could be engineered to more meaningful features.

The difference between features between plan_list_price and the actual_amount_paid by the user gives the discount availed by a user. This feature can affect the future decision of a user more than the given features. Hence the new feature 'Discount' has been engineered. Also, based on the feature discount another feature has been engineering to find out if the user got a discount or not. This feature can help to conclude if providing a discount can improve business or not. Hence it is an interesting feature.

Some other features engineered from the Data Frame Transaction are amount_per_day which is the quotient of features actual_amount_paid and payment_plan_days. The membership_duration which is the difference of features membership_expire_date and transaction_date. In some cases, membership_duration has been found negative and those values were clipped to 0 values. The feature auto renew can affect the churning nature of a user and it was calculated if the user auto-renewed or not in each cases. It is observed that there are some infinite values for amount_per_day. hence have been converted to nans, which are again converted to zeros. Hence new set of more useful and meaningful features have been engineered from the existing features.

The root and intermediate features like 'discount', 'plan_list_price','actual_amount_paid','transaction_date' and 'is_auto_renew' have been removed from the data frame. The final features are payment_method_id, payment_plan_days, membership_expire_date, is_cancel, is_discount, amount_per_day,membership_duration and is_auto_renew. Memory reduction was executed on the data frame to reduce memory of the new frame with new features.

4.1.5.2 Grouping Data

The transaction data frame includes all the transactions carried out by a user so far and hence have multiple rows present for the same user, i.e. ame msno Id. It is necessary to group all the transactions of a single user and convert to one tuple for analysis. From the preprocessed transaction data, the following features have been used to merge the users. The most frequent payment_method_id used by a particular user has been assigned to that user. The mean of payment_plan_days has been assigned to the user because that can extrapolate the average usage of the user in the website every time he makes a payment. This is the average payment plan days used by the member for a given transaction. For the feature not_auto_renew the count of which a particular user has auto-renewed is calculated. It gives the number of times the user has not auto renewed his license. Frequency of each user performing a transaction is calculated using the count of msno which is the unique ID for each user. The feature is_cancel is added over each user and it gives the number of times the user has cancelled his transactions. Also, the feature is_discount is added over each user and it gives the number of times the user has been given a discount by KKBox. The mean of the feature membership_duration gives the average membership duration of the user. Finally, the maximum of the feature Membership_expiration was found out to identify the present expiry date of the membership of the given user.

Hence the final processed data frame Transactions contains each user ID and the following features:

-payment_method_id (Most Frequent Payment method)

-payment_plan_days (Mean of payment plan days the user used everytime he made a transaction)

-not_auto_renew (Number of times a user auto-renewed)

-msno_count (Number of transactions a user performed in the website sofar)

-is_cancel (Number of times a user cancelled a transaction)

-is_discount (Number of times a user availed a discount)

-amount_per_day (The amount per day a user paid when he was on the website)

-membership_duration (Average membership duration of the user)

membership_expire_date (Existing expiry of membership of a user)

4.1.6 Merging Members and Transactions

The frames members and transactions have been merged to get a holistic view of a given user. Since there are users whose information has not been given, a left join of the frame transactions was performed with the frame members.

The categorical variants bd and registered_via were replaced by mode value in the evnets of detection of null values. The numerical variant of bd, in this column the null values have been replaced with the mean of that column. The numerical variant of bd, in the events of null values were replaced by the mean values of the feature.

On combining these two data frames we have two features membership_expire_date (max of expiration date from the transactions data) and the registration_init_time(the time when the user has registered). Difference of these two columns gives the duration of time for which the user is subsribed with KKBox. If we divide with 365. we can get the number of years the user has been with KKBox. The feature was engineered and represented as a new feature long_time_user.

Now the root and intermediate features like 'membership_expire_date', 'registration_ init_time' and 'reg_mem_duration have been removed and memory has been reduced.

The data frame is copied to another dictionary for scaling the features. A minmax scalar is used the data points under features payment_plan_days', 'not_auto_renew', 'msno_ount', 'is_cancel', 'is_discount', 'amount_per_day', 'membership_duration' , 'long_time_user' and 'bd' have been scaled to values between 0 and 1. Also, all the categorical variables have been converted into numerical using label encoding. One hot encoding has not been used since there are many values for the categorical values. Therefore, one hot encoding results in having many columns in the final data frame. Categorical features 'payment_method_id', 'city', 'registered_via' are converted to numerical using label encoder.

Hence the final data set after feature engineering and preprocessing is as below:

	msno	payment_method_id	payment_plan_days
0	+++FOrTS7ab3tlgIh8eWwX4FqRv8w/FoiOuyXsFvphY=	33	0.015556
1	+++IZseRRiQS9aaSkH6cMYU6bGDcxUieAi/tH67sC5s=	20	0.894444
2	+++hVY1rZox/33YtvDgmKA2Frg/2qhkz12B9yICvh8o=	39	0.066667
3	+++I/EXNMLTijfLBa8p2TUVVVp2aFGSuUI/h7mLmthw=	37	0.063810
4	+++snpr7pmobhLKUgSHTv/mpkqgBT0tQJ0zQj6qKrqc=	39	0.064198

ays	not_auto_renew	msno_count	is_cancel	is_discount	amount_per_day	m
	0.004098	0.000000	0.0	0.0	0.000000	0.
	0.008197	0.004115	0.0	0.0	0.554580	0.
	0.000000	0.016461	0.0	0.0	0.435271	0.
	0.000000	0.082305	0.0	0.0	0.620891	0.
	0.000000	0.106996	0.0	0.0	0.630842	0.

day	membership_duration	city	bd	gender	registered_via	bd_c	long_time_user
	0.001382	13	0.200000	2.0	3	0.0	0.545455
	0.121752	4	0.133333	2.0	0	0.0	0.227273
	0.008292	0	0.000000	0.0	2	1.0	0.090909
	0.013570	13	0.177778	1.0	3	0.0	0.272727
	0.008404	0	0.000000	0.0	2	1.0	0.090909

Fig 10: Data frame after data pre-processing

4.1.7 Merging Final Processed Data Frame with Train and Test for Analysis

Finally, the processed data frame is merged with training and testing data separately based on the user ID which is msno.

A pair wise correlation of features (columns) has been performed and a heatmap and correlation table has been generated. The results are as below:

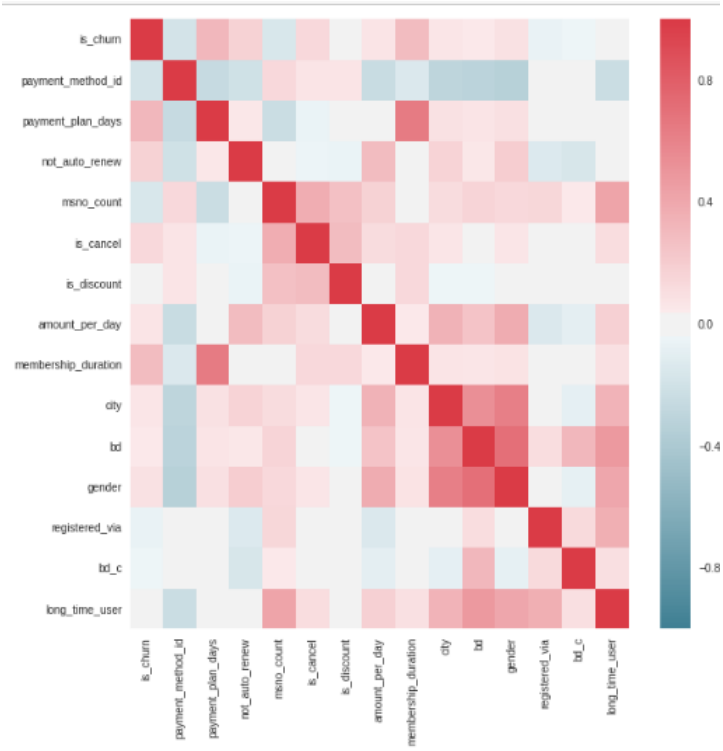


Fig 11: Heatmap of pairwise correlation between features.

From the correlation Heatmap and Table (APPENDIX I) it has been overserved that there is high negative correlation between is_churn and payment_id and msno_count. Also, there is a high positive correlation between is_churn and payment_plan_days, not_auto_renew, is_cancel and membership_duration.

4.1.8 Splitting into Test Data and Train Data using Under Sampling

As it can be seen from Fig 1 there is a huge class imbalance between class 0 and class 1. Therefore, under sampling has been done, to reduce the class imbalance. The size of class 0 is 5 times that of class 1. i.e. Every time almost one user churns when five users continue.

4.1.9 Modelling

Several Modellings have been executed and results have been analyzed. The details are presented in APPENDIX II. The best Model turned out to be Random Forest and Grid Search was performed to analyze its best parameters.

4.1.10 Grid Search, Cross Validation and Random Forest

GridSearch as well Cross validation are used to find out the best parameters. Also since there is a class imbalance, Ensemble methods such as random forest and Gradient boosting have been used. The data was tested with various models (Presented in APPENDIX II) and the best accuracy was evaluated. The best model is presented here.

Some parameters have been given to Random forest classifier to identify the best one. The grid search returns the best parameters. This was

performed several times and with different parameters and the best parameters turned out as

```
RandomForestClassifier(bootstrap=True, class_weight=None, criterion='gini',
            max_depth=None, max_features='log2', max_leaf_nodes=None,
            min_impurity_split=1e-07, min_samples_leaf=1,
            min_samples_split=2, min_weight_fraction_leaf=0.0,
            n_estimators=20, n_jobs=1, oob_score=False, random_state=None,
            verbose=0, warm_start=False)
```

Fig 12: Random Forest best Parameters after Grid Search

4.2 Results

4.2.1 Precision- Recall

The Results have been evaluated using precision, Recall, F1 score and support.

```
: print(classification_report(y_test,t_pred))
```

	precision	recall	f1-score	support
0	0.98	0.95	0.96	543775
1	0.53	0.71	0.61	45393
avg / total	0.94	0.93	0.93	589168

Fig 13: Precision- Recall-F1-score-Support

Precision was found as 0.94 and recall 0.93. The F1-score value is 0.93.

4.2.2 ROC curve

ROC curve was plotted to analyze the results better.

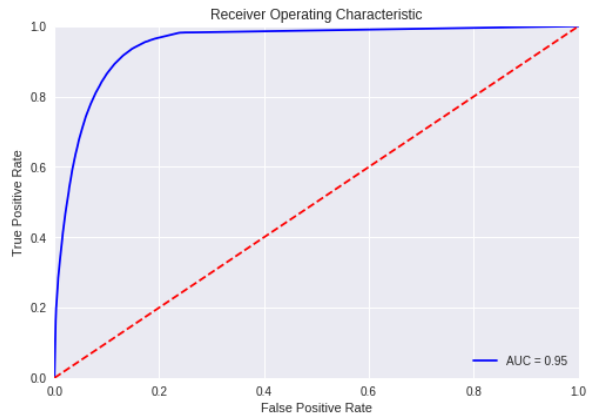


Fig 14: ROC curve

The area under ROC curve for the best model is 0.95.

The results have been satisfactory and in-line with the initial assumptions.

4.3 Discussion

Our best entry was placed in 115-th position among around 500 other submissions.

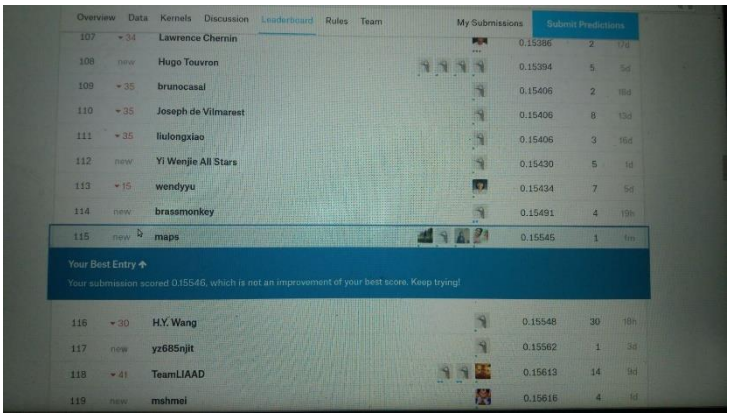


Fig 15: Position in Leaderboard for the best submission.

5. Future Work

One major consideration here is the usefulness of the User_logs file. The data has millions of records about each user's listening habits and the file size is huge. The usefulness of this data was doubtful after analysis because of the presence of null values and incomplete information about a lot of users. A suggested future work would be normalizing and scaling this data log to find out if there is any correlation between listening habit and duration of time of a single song a user listens to, on an average basis.

6. Conclusion

In this project we aimed at developing a machine learning algorithm to accurately predict if the given user will churn a subscription or not. It is performed using the donated dataset from KKBOX for Kaggle challenge "WSDM - KKBox's Churn Prediction Challenge -Can you predict when subscribers will churn?". From the initial data analysis, we concluded that there are lots of features like a users' age, gender, payment method used, registration method used, history of listening songs, cancellation history, renewal history, city etc. After analyzing and preprocessing the data new features were engineered and it was tested on strong ensemble techniques like Random Forest, Gradient boosting and Neural Networks The best model suggested predicted the test data with an accuracy of 0.94. This method is much better than the present way of survival analysis techniques to determine the residual membership life time for each subscriber.

Bibliography

[1] Machine Learning, Tom Mitchell, McGraw Hill, 1997.

[2] http://pandas.pydata.org/pandas-docs/stable/

[3] https://www.kaggle.com/headsortails/should-i-stay-or-should-i-go-kkbox-eda

[4] https://www.kaggle.com/the1owl/regressing-during-insomnia-0-21496

[5] https://www.kaggle.com/jeru666/memory-reduction-and-data-insights

[6] https://www.analyticsvidhya.com/blog/2017/03/imbalanced-classification-problem/

[7] http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

[8] http://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html

[9] http://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html

[10] http://scikit-learn.org/stable/modules/generated/sklearn.metrics.roc_curve.html

[11] http://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.LabelEncoder.html

[12] https://docs.scipy.org/doc/numpy/

APPENDIX I: Pair Correlation Table

	is_churn	payment_method_id	payment_plan_days	not_auto_renew	msno_count	is_cancel	is_discount	amount_per_day	membership_ duration	city	bd	gender	registered_vis	bd_c	long_time_user
is_churn	1	-0.187939508	0.31412881	0.172895906	-0.16399065	0.13447709	0.008633793	0.070560687	0.295386201	0.069256956	0.048589	0.088017	-0.065921921	-0.045156293	-0.002299436
payment_method_id	-0.187939508	1	-0.262687949	-0.207015558	0.137719501	0.073058551	0.074640776	-0.244446191	-0.145430538	-0.301722461	-0.31703	-0.33614	-0.022417477	0.020417853	-0.23653528
payment_plan_days	0.31412881	-0.262687949	1	0.062250267	-0.238430419	-0.055414694	-0.030304803	0.023321484	0.643333212	0.086874589	0.076751	0.095427	-0.017456237	-0.018962634	0.032321798
not_auto_renew	0.172895906	-0.207015558	0.062250267	1	-0.027795184	-0.047761788	-0.057805393	0.291857398	0.035958014	0.165277059	0.055616	0.199984	-0.139402287	-0.17045997	-0.002900513
msno_count	-0.16399065	0.137719501	-0.238430419	-0.027795184	1	0.373682935	0.27147437	0.175468677	0.023335602	0.121906239	0.162987	0.137285	0.142895942	0.054649705	0.423808499
is_cancel	0.13447709	0.073058551	-0.055414694	-0.047761788	0.373682935	1	0.290013934	0.117723099	0.136209355	0.068006815	0.032335	0.066181	0.015279655	-0.036074657	0.111147445
is_discount	0.008633793	0.074640776	-0.030304803	-0.057805393	0.27147437	0.290013934	1	0.006563993	0.140365615	-0.041749795	-0.04533	-0.03261	0.005275247	0.005235838	0.023099223
amount_per_day	0.070560687	-0.244446191	0.023321484	0.291857398	0.175468677	0.117723099	0.006563993	1	0.053403597	0.345420213	0.251385	0.376904	-0.140989978	-0.09593002	0.18282556
membership_duration	0.295386201	-0.145430538	0.643333212	0.035958014	0.023335602	0.136209355	0.140365615	0.053403597	1	0.074202261	0.069415	0.083741	0.001273707	-0.004033013	0.100970986
city	0.069256956	-0.301722461	0.086874589	0.165277059	0.121906239	0.068006815	-0.041749795	0.345420213	0.074202261	1	0.536297	0.62061	0.022415548	-0.08892606	0.343506124
bd	0.048588526	-0.317026348	0.07675088	0.055616322	0.162987336	0.032334768	-0.045327522	0.251384973	0.069415495	0.536297259	1	0.708026	0.110397015	0.317849815	0.48405014
gender	0.088016897	-0.33614082	0.095426782	0.199983648	0.137285174	0.066181484	-0.032613438	0.376903894	0.083741422	0.620609851	0.708026	1	0.019018643	-0.083151087	0.411928024
registered_vis	-0.065921921	-0.022417477	-0.017456237	-0.139402287	0.142895942	0.015279655	0.005275247	-0.140989978	0.001273707	0.022415548	0.110397	0.019019	1	0.129579371	0.365636309
bd_c	-0.045156293	0.020417853	-0.018962634	-0.17045997	0.054649705	-0.036074657	0.005235838	-0.09593002	-0.004033013	-0.08892606	0.31785	-0.08315	0.129579371	1	0.108720736
long_time_user	-0.002299436	-0.23653528	0.032321798	-0.002900513	0.423808499	0.111147445	0.023099223	0.18282556	0.100970986	0.343506124	0.48405	0.411928	0.365636309	0.108720736	1

APPENDIX II Test Results and log of Other Models

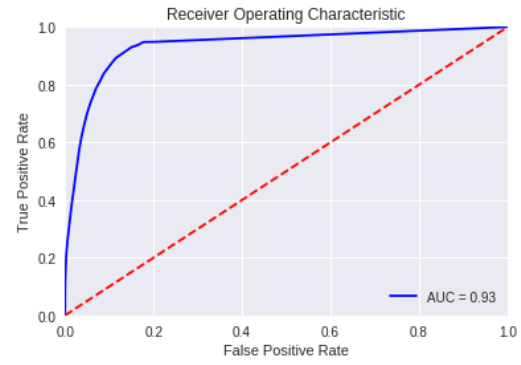
1. Splitting into training and testing data. No Scaling.

Random forest was implemented using grid search and cross validation without performing the splitting and scaling of testing data.

Results:

```
print(classification_report(y_test,t_pred))
```

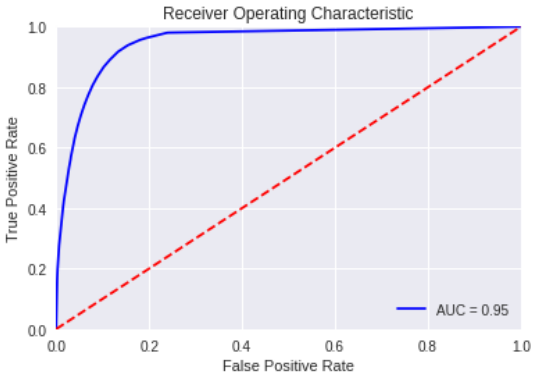
	precision	recall	f1-score	support
0	0.96	0.97	0.97	543775
1	0.62	0.51	0.56	45393
avg / total	0.93	0.94	0.94	589168



2. Under Sampling without Scaling

```
print(classification_report(y_test,t_pred))
```

	precision	recall	f1-score	support
0	0.97	0.95	0.96	543775
1	0.53	0.71	0.60	45393
avg / total	0.94	0.93	0.93	589168



3. Logs of parameter testing for Random Forest and Results

3.1 Default - no sub sampling. No scaling

I trial Criterion = entropy. No subsampling, no scaling.

	precision	recall	f1-score	support
0	0.96	0.97	0.97	598460
1	0.62	0.50	0.55	49625
avg / total	0.93	0.94	0.94	648085

II trial Criterion = entropy. No subsampling, no scaling, n_estimators = 100

	precision	recall	f1-score	support
0	0.96	0.97	0.97	598460
1	0.62	0.52	0.56	49625
avg / total	0.93	0.94	0.94	648085

III trial Criterion = 'gini', No subsampling, no scaling, n_estimators = 10, max_features = 'log2',

	precision	recall	f1-score	support
0	0.96	0.97	0.97	598460
1	0.62	0.50	0.55	49625
avg / total	0.93	0.94	0.94	648085

IV trial Criterion = 'gini', Undersampling, no scaling., n_estimators = 10, max_features = 'log2'

	precision	recall	f1-score	support
0	0.99	0.87	0.93	598460
1	0.37	0.91	0.53	49625
avg / total	0.94	0.88	0.90	648085

V trial criterion = 'gini', n_estimators = 100, min_samples_leaf = 0.01

	precision	recall	f1-score	support
0	1.00	0.68	0.81	598460
1	0.20	0.97	0.33	49625
avg / total	0.94	0.70	0.77	648085

VI trial Criterion = 'gini' n_estimators = 10, max_features = 'log2', here bd is not categorical variable.

	precision	recall	f1-score	support
0	0.96	0.97	0.97	543775
1	0.62	0.50	0.55	45393
avg / total	0.93	0.94	0.93	589168

VII trial Under sampling, Sample size of 0 is 5 times, n_estimators = 10, criterion = 'gini' (here bd is not categorical variable.)

	precision	recall	f1-score	support
0	0.97	0.95	0.96	543775
1	0.53	0.70	0.60	45393
avg / total	0.94	0.93	0.93	589168

VIII trial After Scaling. No under sampling. n_estimators = 10, criterion = 'gini'

	precision	recall	f1-score	support
0	0.96	0.97	0.97	543775
1	0.62	0.50	0.55	45393
avg / total	0.93	0.94	0.93	589168

IX trial Under sampling. Sample size of 0 is 5 times, n_estimators = 20, criterion = 'gini', max_features='log2').

	precision	recall	f1-score	support
0	0.98	0.95	0.96	543775
1	0.53	0.71	0.61	45393
avg / total	0.94	0.93	0.93	589168

X trial n_estimators = 10, criterion = 'gini', Filtered columns based on importance.

	precision	recall	f1-score	support
0	0.96	0.98	0.97	543775
1	0.65	0.47	0.54	45393
avg / total	0.93	0.94	0.93	589168

XI trial n_estimators = 50, criterion = 'gini', max_features = 'log2',
Filtered columns based on importance.

	precision	recall	f1-score	support
0	0.96	0.98	0.97	543775
1	0.65	0.46	0.54	45393
avg / total	0.93	0.94	0.93	589168

XII trial n_estimators = 50, criterion = 'gini', max_features = 'log2',
Filtered columns based on importance.

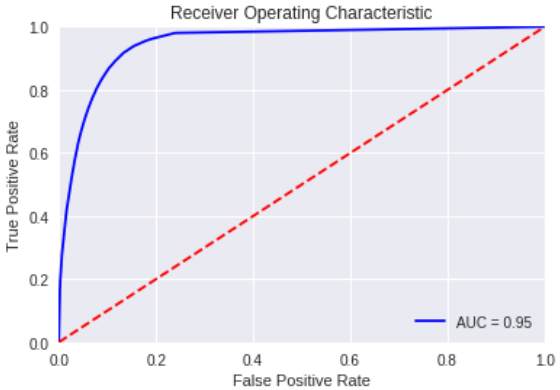
	precision	recall	f1-score	support
0	0.97	0.95	0.96	543775
1	0.51	0.67	0.58	45393
avg / total	0.94	0.93	0.93	589168

XIII trial n_estimators = 50, criterion = 'gini', max_features = 'log2', max_depth=10
Filtered columns based on importance.

	precision	recall	f1-score	support
0	0.96	0.97	0.97	543775
1	0.61	0.55	0.58	45393
avg / total	0.94	0.94	0.94	589168

4. Gradient Boosting – Testing and Results

	precision	recall	f1-score	support
0	0.95	0.98	0.97	543775
1	0.65	0.35	0.45	45393
avg / total	0.93	0.94	0.93	589168



5. Logs of parameter testing Gradient Boosting and Results

I trial default values, no sub sampling, scaling.

	precision	recall	f1-score	support
0	0.95	0.99	0.97	543775
1	0.75	0.36	0.49	45393
avg / total	0.93	0.94	0.93	589168

II trial default values, sub sampling, scaling. Sample size of 0 is 5 times.

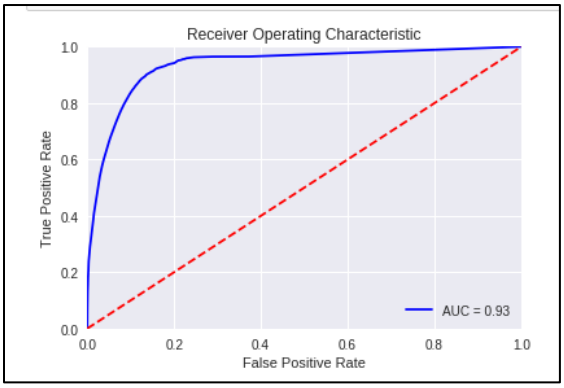
	precision	recall	f1-score	support
0	0.96	0.97	0.97	543775
1	0.59	0.55	0.57	45393
avg / total	0.93	0.94	0.93	589168

III trial learning_rate=0.01, n_estimators=200, subsample = 0.2, sample size of 5 times.

	precision	recall	f1-score	support
0	0.95	0.98	0.97	543775
1	0.66	0.35	0.46	45393
avg / total	0.93	0.94	0.93	589168

6. Artificial Neural Networks – Testing and Results

	precision	recall	f1-score	support
0	0.97	0.96	0.96	543775
1	0.54	0.58	0.56	45393
avg / total	0.93	0.93	0.93	589168



7. Logs of parameter testing Neural Networks and Results

I trial solver='sgd', alpha=1e-4, hidden_layer_sizes = (40,40,40), random_state=1, activation = 'relu', max_iter = 400
No under sampling, scaling.

	precision	recall	f1-score	support
0	0.93	1.00	0.96	543775
1	0.75	0.14	0.24	45393
avg / total	0.92	0.93	0.91	589168

II trial solver='sgd', alpha=1e-4, hidden_layer_sizes=(40,40,40), random_state=1, activation = 'relu', max_iter = 100
No undersampling, scaling.

	precision	recall	f1-score	support
0	0.93	1.00	0.96	543775
1	0.75	0.14	0.24	45393
avg / total	0.92	0.93	0.91	589168

III trial solver='sgd', alpha=1e-4, hidden_layer_sizes=(40,40,40),random_state=1,activation = 'relu', max_iter = 100

	precision	recall	f1-score	support
0	0.95	0.95	0.95	543775
1	0.40	0.37	0.39	45393
avg / total	0.91	0.91	0.91	589168

IV trial solver='sgd', alpha=1e-4, hidden_layer_sizes=(10,10,10),random_state=1,activation = 'relu', max_iter = 100, learning_rate = 'adaptive'

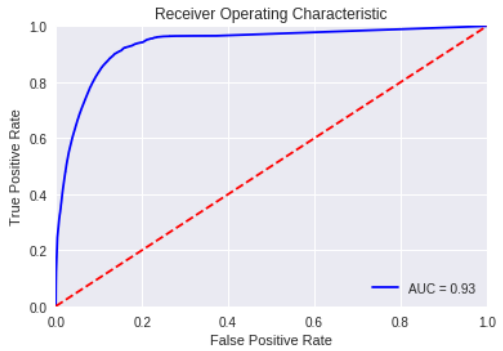
	precision	recall	f1-score	support
0	0.94	0.97	0.96	543775
1	0.50	0.32	0.39	45393
avg / total	0.91	0.92	0.91	589168

8. Random forest based on feature importance

On analyzing the features, it was observed some features ('payment_method_id','payment_plan_days','not_auto_renew','msno_count','is_cancel','membership_duration') have high correlation with is_churn. A random forest model was executed based on these features alone.

The results are

	precision	recall	f1-score	support
0	0.96	0.98	0.97	543775
1	0.65	0.46	0.54	45393
avg / total	0.93	0.94	0.93	589168



9. Random forest with feature importance and under sampling

Above model was combined with under-sampling to check for improvement of results

The results are:

	precision	recall	f1-score	support
0	0.96	0.97	0.97	543775
1	0.61	0.54	0.57	45393
avg / total	0.93	0.94	0.94	589168