Predicting Churn to Minimize Business Loss

Contents

1 – Introduction:

What is churn and why is it of value to businesses:

Customer churn, also known as customer attrition, occurs when customers stop doing business with a company. The companies are interested in identifying segments of these customers because the price for acquiring a new customer is usually higher than retaining the old one. For example, if Netflix knew a segment of customers who were at risk of churning they could proactively engage them with special offers instead of simply losing them.

Churn can also be defined as "The use of customer data and/or feedback to forecast the likelihood of a customer or group of customers discontinuing their subscription in the future."

Why Is It Necessary?

Having the ability to accurately predict future churn rates is necessary because it helps your business gain a better understanding of future expected revenue.

In addition, when you're able to use churn prediction to forecast the potential churn rate of a particular customer, it allows you to target that individual in an attempt to prevent them from discontinuing their subscription with you.

And, since the cost of acquiring a new customer is 5x higher than keeping an existing one, there's plenty of revenue-based reason to do everything in your power to keep those existing customers.

Predicting churn rates can also help your business identify and improve upon areas where customer service is lacking. And, by

making those improvements, you can decrease churn and improve revenue numbers.

In the end, the bottom line is that churn prediction is essential because it helps you understand what preventative steps are necessary to ensure lost revenue is minimized.

Where Is It Used?

Churn prediction is used in a variety of different industries and types of businesses.

It is, however, most relevant to SaaS companies and membership based businesses that charge an ongoing monthly, quarterly, or annual fee for their software or services.

As far as how churn prediction can be used within your business, it's one of the key components of determining the lifetime value of customers. And, armed with accurate, real-time data about the lifetime value of your customers, your company will be in a much better position to ensure that you're making decisions that keep you moving forward.

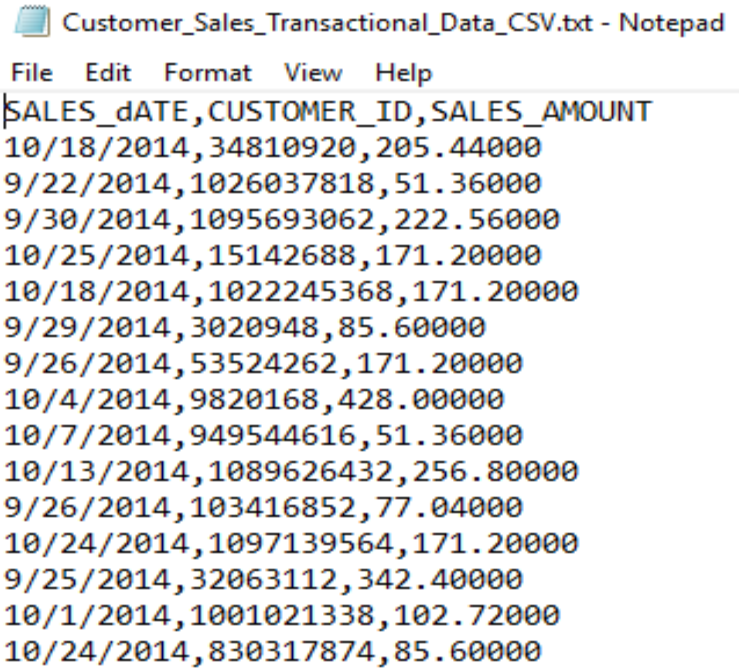
So, this project is aimed to leverage customer's data of sales of Chicago metropolitan area and predict whether he/she will churn in a given time or will remain loyal to the business!

2 - Description of the Data:

The data I will be using in this project is of a Mall located in Chicago's multiple cities consisting of customer's sales data. First let's have a look at how our data looks like:

Our data is in text form but it is not text data rather, it is numerical data stored in text file, so we first load data into jupyter environment a sav e a copy of it to a dataframe in order to manipulate it.

Data before reading:



Data After reading into Dataframe:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import datetime
```

```
data=pd.read_csv('Customer_Sales_Transactional_Data_CSV.txt')
```

```
df=data.copy()
```

```
df.head()
```

	SALES_dATE	CUSTOMER_ID	SALES_AMOUNT
0	10/18/2014	34810920	205.44
1	9/22/2014	1026037818	51.36
2	9/30/2014	1095693062	222.56
3	10/25/2014	15142688	171.20
4	10/18/2014	1022245368	171.20

So, we can see that our data initially contains 03 columns: 'SALES_dATE',' CUSTOMER_ID' and 'SALES_AMOUNT'. So given a customer_id, we know his/her sales date and sales amount and there are more than one rows against one customer id.

Size of Data:

Our data contains more than one million rows.

```
: df.shape

: (1058198, 3)
```

This is a six week sales data of different customers. The problem of churn prediction will be solved by this data in such a way that first five weeks' data will be used to train the model and sixth week data will be used to test or predict churn against a given customer.

Feature Engineering:

Since our data consists of only 03 columns. We need to do feature engineering to get better insight from our data.

Adding location from FourSquare API for Chicago area:

I used Foursquare API to add location to given data and then predict churn with respect to area.

This will help in focusing on areas with greater churn and hence that particular mall.

```
df["area"] = np.random.choice(chicago_areas, size=len(df))
```

```
df.head()
```

	SALES_dATE	CUSTOMER_ID	SALES_AMOUNT	area
0	10/18/2014	34810920	205.44	Genoa City, WI-IL
1	9/22/2014	1026037818	51.36	Michigan City-LaPorte, IN-MI
2	9/30/2014	1095693062	222.56	Michigan City-LaPorte, IN-MI
3	10/25/2014	15142688	171.20	Chicago-Aurora-Elgin-Joliet-Waukegan
4	10/18/2014	1022245368	171.20	Round Lake Beach-McHenry-Grayslake

Now, we need to split date column into month, day and date plus week number of the year [0-52] respectively like:

```
weeks=[]
for i in range(1058198):
    weeks.append(datetime.date(df.loc[i,'Year'],df.loc[i,'Month'],df.loc[i,'Day']).isocalendar()[1])
```

```
df['weeks']=weeks
```

```
df.head()
```

	SalesDate	CoustomerId	SalesAmount	Year	Day	Month	weeks
0	10/18/2014	34810920	205.44	2014	18	10	42
1	9/22/2014	1026037818	51.36	2014	22	9	39
2	9/30/2014	1095693062	222.56	2014	30	9	40
3	10/25/2014	15142688	171.20	2014	25	10	43
4	10/18/2014	1022245368	171.20	2014	18	10	42

Next thing to do is to find maximum, minimum and average amount of sale against each customer.

```
Week_0_5.columns=['CustomerId','Mean_1','Max_1','Min_1','Count_1','StandardDeviation_1','Revenue_1','Mean_2','Max_2','Min_2'
```

```
Week_0_5.fillna(0,inplace=True)
```

```
Week_0_5['TotalRevenue']=Week_0_5['Revenue_1'] + Week_0_5['Revenue_2'] + Week_0_5['Revenue_3'] + Week_0_5['Revenue_4'] + Wee
```

```
Week_0_5.head()
```

	CustomerId	Mean_1	Max_1	Min_1	Count_1	StandardDeviation_1	Revenue_1	Mean_2	Max_2	Min_2	...	Count_4	StandardDeviation_4	Re
0	7260	171.20	171.20	171.20	1.0	0.0	171.20	128.4	85.60	171.20	...	3.0	19.768473	
1	11024	68.48	68.48	68.48	1.0	0.0	68.48	85.6	51.36	119.84	...	1.0	0.000000	
2	12476	51.36	51.36	51.36	1.0	0.0	51.36	0.0	0.00	0.00	...	1.0	0.000000	
3	14156	171.20	171.20	171.20	1.0	0.0	171.20	171.2	171.20	171.20	...	1.0	0.000000	
4	17606	171.20	171.20	171.20	1.0	0.0	171.20	0.0	0.00	0.00	...	0.0	0.000000	

5 rows × 32 columns

And similarly other columns are also introduced which is all present in jupyter notebook

3 – Methodology:

This section aims at describing the methodology used to predict churn effectively.

The analysis of data begins by doing some feature engineering. Few new columns are added to data and none is removed at this stage. Data contains 03 columns at first. We start by splitting date

columns into its respective columns i.e. date, day, month and week number respectively.

Then we add respective Chicago area to each of the sale record randomly since the purpose is to learn and apply.

So, at this stage our data has a total of 08 columns with the last one being 'area' representing area in which sale was performed.

As mentioned in the start of the report, our data is of six week sales done by different customers in different areas of Chicago in same superstore. So, next step is that we separate data week wise. We will now have data of week 1 – week 6 each. As shown:

```
Week_1=df[df['weeks']==38]
Week_1=Week_1.groupby('CoustomerId')['SalesAmount'].agg({'Statistical Measures':[np.mean,np.min,np.max,np.count_nonzero,np.st
Week_1.columns=['Mean','Max','Min','Count','StandardDeviation','Total']
Week_1=Week_1.reset_index()
```

Similarly, for all other subsequent weeks.

We plan to use data of first five weeks for training purpose and last week (week 06) data for testing so, we merge data of first five weeks while keeping data of sixth data separate.

We also remove Nan from our data using numpy 'np. fillna()' function.

The role of Foursquare was to give us Chicago locations and then we assigned it randomly to our data transactions.

We also added all revenue from week 01 till week 05 and assigned it to the column named 'Total Revenue'.

Now our model is ready to be fed to the model.

But our 'Area' column is categorical so we convert it into numerical using one-hot encoding capability of python numpy library.

Finally, we also normalized our data using standardscaler () function of sklearn. preprocessing

So, let's discuss ML part of the project.

We used different ML algorithms for our data to get the best response. Following algorithms were used:

- KNN
- Decision Tree
- Gaussian NB
- Logistic Regression

Logistic Regression:

KNN gav e us the following results.

```
  precision    recall  f1-score   support

      False       1.00      1.00      1.00     33181
       True       1.00      1.00      1.00     24372

avg / total       1.00      1.00      1.00     57553

accuracy_score(ytest,lr_pred)

0.9999652494222716
```

KNN:

```
precision    recall  f1-score   support

      False       1.00      1.00      1.00     33181
       True       1.00      1.00      1.00     24372

avg / total       1.00      1.00      1.00     57553


accuracy_score(ytest,lr_pred)

0.9999652494222716
```

Decision Tree:

```
accuracy_score(ytest,lr_pred)
```

```
0.88989934762
```

Gaussian NB:

```
accuracy_score(ytest,lr_pred)

0.859247624331
```

4 – Results:

From our results we conclude that very few customers will churn so this particular superstore need not to worry about any of its superstore in Chicago. It should be confident about loyalty of its customers. The customers which are likely to churn as per our model needs special attention of superstore in order to make them keep visiting superstore and do transactions.

5 – Conclusion:

We conclude our project by observing that superstore is on safe side and its customers of all areas are not going to churn in this given time of one week and this superstore needs not to give any promotional packages to the customers since they will remain loyal as per our model predictions