

Consumer Churn Program

Framework, capabilities and lessons learned (well, at least so far....)

What is Churn Rate?

- [Wikipedia:](#)

“**Churn** rate (sometimes called attrition rate), in its broadest sense, is a measure of the number of individuals or items moving out of a collective group over a specific period of time”

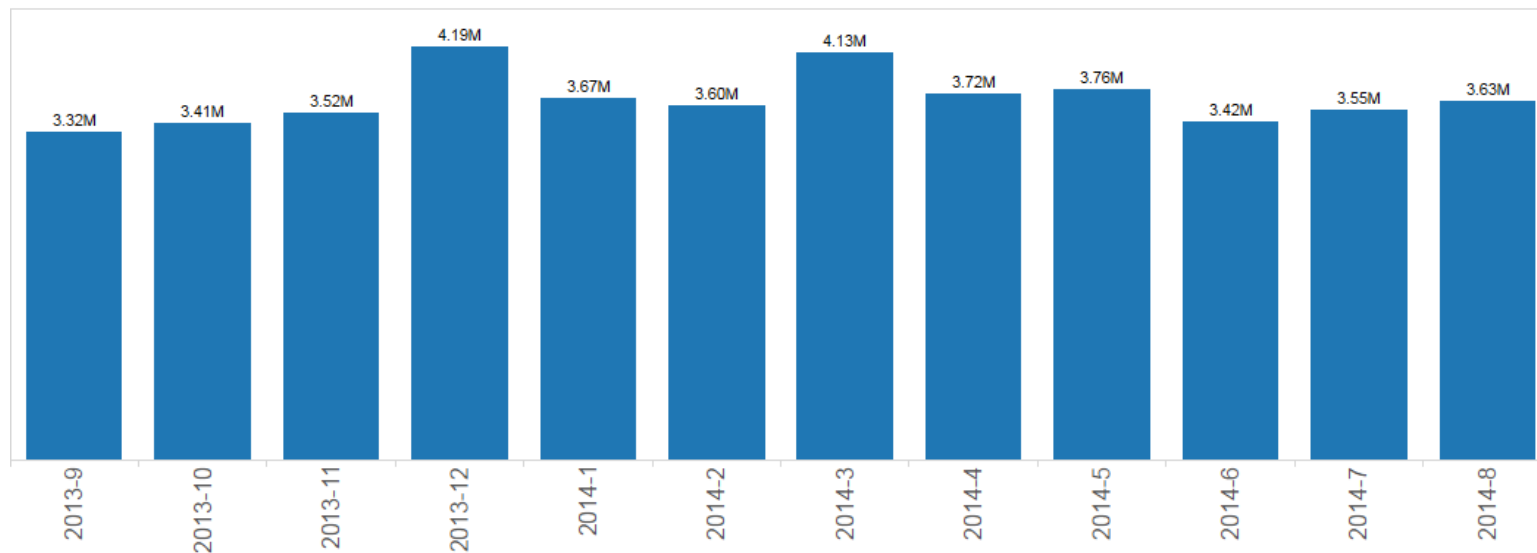
= **Customer leaving**



Current Churn impact @ PayPal

Why Churn ?

Churn Volume



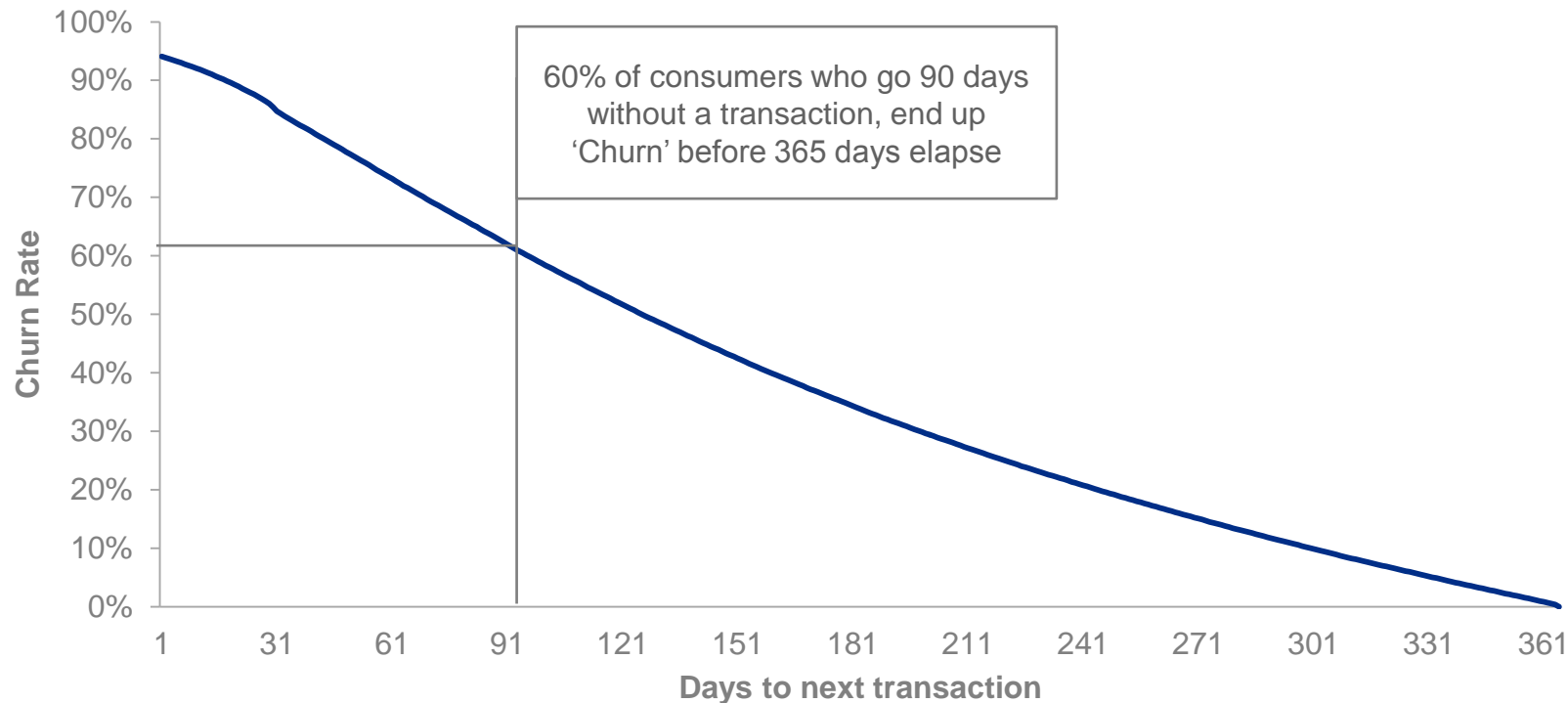
Annually....

Churn rate is very widely discussed topic and plays a very significant role in PayPal financial growth and as it is often more expensive to obtain new customers than to keep existing customers on board.

43.93M

Assumption for a consumer churn

From 30,000 feet up



Before and after...

The thinking around churn

- Wait, the consumer hasn't churned yet, we'll do xx after they churn
 - Churn happens when we found out
- Let's assign a probability periodically (monthly) and figure out today, if someone's going to churn next year. It's ok if you're not super accurate
 - A consumer churned on the day of their last transaction, not when we found out, but, when they did their last transaction (probably)

Rough idea of end product

What do we think will resonate with our internal customers

Consumers: Churn Probability - Region: NA - Country: CANADA




Predicted Churn Date	Consumer ID (dim_customer.cust_id)	Feature Segment	
9/1/2014	1161073680815551472	daysSinceLastTxn, mthOfLastTxn, txnsPerDay	0.9420
	1163484604045261769	daysSinceLastTxn, avgDateDiff, mthOfLastTxn	0.8980
	1164010385686670617	avgDateDiff, daysSinceLastTxn, maxDateDiff	0.9500
	1168522338607095666	avgDateDiff, daysSinceLastTxn, maxDateDiff	0.9560
	1169818400784242809	daysSinceLastTxn, avgDateDiff, maxDateDiff	0.9360
	1176727687991066529	daysSinceLastTxn, avgDateDiff, spendPerDay	0.9360
	1181281662193280439	daysSinceLastTxn, maxDateDiff, mthOfLastTxn	0.9260
	1183665695078895704	avgDateDiff, daysSinceLastTxn, maxDateDiff	0.9020
	1187385624820871550	avgDateDiff, daysSinceLastTxn, maxDateDiff	0.8260
	1188318310880402734	daysSinceLastTxn, mthOfLastTxn, txnsPerDay	0.9420
	1192610902218077779	daysSinceLastTxn, avgDateDiff, maxDateDiff	
	1193274790258224511	txnsPerDay, daysSinceLastTxn, maxDateDiff	
	1195701701950825182	daysSinceLastTxn, spendPerDay, avgDateDiff	
	1199332100883009392	daysSinceLastTxn, mthOfLastTxn, txnsPerDay	
	1200319129973926008	daysSinceLastTxn, avgDateDiff, maxDateDiff	
	1212992953473438973	avgDateDiff, daysSinceLastTxn, maxDateDiff	
	1214349417779392369	avgDateDiff, daysSinceLastTxn, maxDateDiff	
	1215450039683340908	daysSinceLastTxn, mthOfLastTxn, txnsPerDay	
	1218270749336777162	avgDateDiff, daysSinceLastTxn, maxDateDiff	
	121830205379061521	daysSinceLastTxn, maxDateDiff, avgDateDiff	

Consumers with highest increases - Region: NA - Country: CANADA

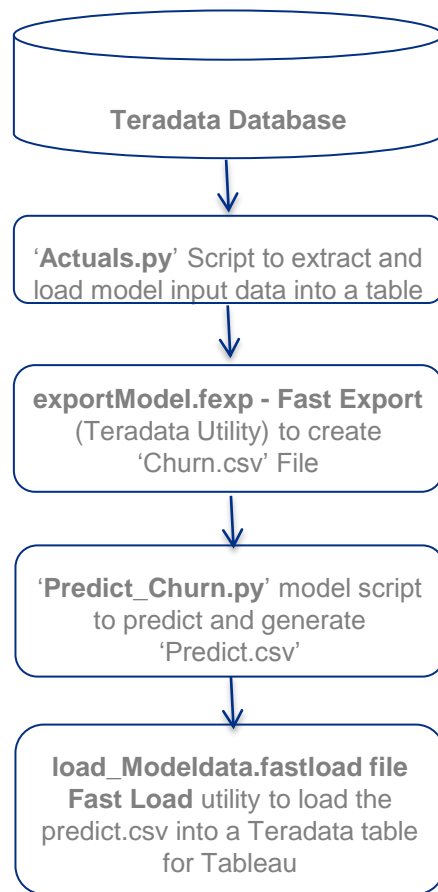
Consumer ID (dim_customer.cust_id)	August 17, 2015	September 6, 2015
1152970086380509619	0.8900	
1153130737007681050		0.9500
1153157356027014241		0.9560
1153163558448296524		0.8940
1153204689093667213	0.9520	
1153299216005542825	0.8700	
1153369885248971772		0.9520
1153436048880516204		0.9340

Predictive Model implementation Exercise

Problem Statement to Final Product

Acquiring Data and EDA	Modeling	Production
		
<ul style="list-style-type: none">Acquiring Data, Feature identification and reduction	<ul style="list-style-type: none">Classification Model – Used Random Forest (for now)	<ul style="list-style-type: none">MVP for time/accuracy and iterate
<ul style="list-style-type: none">Tools Used - Python and Teradata SQL, Teradata Utilities	<ul style="list-style-type: none">import scikitlearn, pandas, numpy, Python RF (Random Forest)	<ul style="list-style-type: none">Python, FLOAD, FEXP, Tableau
<ul style="list-style-type: none">Validate sample size, go multi processing early, QC your data	<ul style="list-style-type: none">Ensemble modelsUse n_folds cross validation scoreAUC to set threshold (need to understand this concept better to tune the algorithm)Focus on Confusion matrix variables like accuracy, recall, precision.	

Acquiring data



Acquiring data

```
gatherDates(startDate, endDate)
daysBetweenTxns ()
createIntermediateTable ()
addToIntermediateTable ()
lastAction ()
lastTxn ()
acctInfo ()
createFinalPredictionTable (eachRun)
moveFinalTables (eachRun)
cleanUp ()
```

```
def gatherDates(startDate, endDate):
    tdSql = """ create multiset table pp_scratch.RP_max_date as
    (
        select tran_customer_id, max(transaction_created_date) as maxTxnDate
        from pp_access_views.dw_payment_sent
        where transaction_status = 'S'
        and tran_customer_id in
        (
            select cust_id
            from pp_discovery_views.dim_customer
            where is_guest_y_n = 'N'
            and guest_wax_user_y_n = 'N'
            and open_wax_acct_y_n = 'N'
            and cust_acct_clsfn_key = 1
        )
        and transaction_created_date between date '%s' and date '%s'
    ) group by 1
    ) with data primary index(tran_customer_id, maxTxnDate)""" % (startDate, endDate)
    runQuery(tdSql, drop="Y", tableName="RP_max_date", exitFlag="Y", clean="Y")

    tdSql = """ create multiset table pp_scratch.RP_min_date as
    (
        select tran_customer_id, min(transaction_created_date) as minTxnDate
        from pp_access_views.dw_payment_sent
        where transaction_created_date <= date '%s'
        and transaction_status = 'S'
        and tran_customer_id in
        (
            select cust_id
            from pp_discovery_views.dim_customer
            where is_guest_y_n = 'N'
            and guest_wax_user_y_n = 'N'
            and open_wax_acct_y_n = 'N'
            and cust_acct_clsfn_key = 1
        )
    ) group by 1
    ) with data primary index(tran_customer_id, minTxnDate)""" % endDate
    runQuery(tdSql, drop="Y", tableName="RP_min_date", exitFlag="Y", clean="Y")
```

Features/Columns (Total 42 features for now) Sample Size = 25K

Column Name	Data Type	Description	Transformed into
cust_id	Char	PayPal Customer ID	
tenureDaysBucket	integer	Days since a consumer opened an account from pp_discovery_views.dim_customer.acct_cre_dt. Binned into an over-1-year (365+) bucket and less-than-1-year (0-365) bucket.	Over 1-year = 1 Less-Than 1-year = 0
txnsPerDay	integer	Avg. Txns per Day	
spendPerDay	integer	Avg. Dollar spend per day	
breadthBucket	integer	Breadth is binned into 3 groups, consumers that have a breadth between 1 and 9 in the bucket '1-9', between 10 and 18 in the bucket '10-18', and the bucket '18+' for consumers with breadth greater than 18.	1-9 = 0 10-18 = 1 18+ = 2
nbrTxns	integer	# of Transactions made by a customer	
Txn Amt	integer	Total Dolar Amount from # of Transactions	
mthOfLastTxn	integer	Last Transaction Month	
nbrRiskEvents	integer	# of reported Suspicious Activity	
last_sent_pmt_txn_status	integer	Last Sent transaction status (Success, Declined)	Success = 0 Declined = 0
last_sent_pmt_txn_usd_amt	integer	Last Sent Transaction Amount	
last_rcv_pmt_txn_status	integer	last Received Transaction Status	Success = 0 Declined = 0
last_rcv_pmt_txn_usd_amt	integer	Last Received Transaction Amount	
tot_bal_equiv_usd_amt	integer	PayPal Balance Amount	



Exploratory Data Analysis

Steps followed

- **Feature Identification and Engineering**
 - **Transaction variables** – high
 - **FPTI variables** - moderate
 - **Demographic** – less
- **Pre Processing the Data**
 - Renaming Features
 - Inconsistent formatting and datatypes
 - Identifying/Removing less significant properties
- Ran a very basic correlations to remove variables that exhibit obvious relationship
- **Quality control**
 - Bugs – did I get what I wanted to get?
 - Conceptual – is this what I really want?

Modelling

Actual Data Science process

- **Model - Classification Model - Random Forest (Null Accuracy(score to beat) – 0.65)**
- **Steps**
 - import pandas, numpy, seaborn (for Heatmap) , matplotlib.pyplot (for basic visulas)
 - from sklearn.ensemble import RandomForestClassifier, from sklearn.preprocessing import StandardScaler
 - from sklearn.cross_validation import Kfold
 - from sklearn.metrics import accuracy_score, recall_score, confusion_matrix
 - Read csv and create dataframe.(both Training and test data sets)
 - Training = 75 K samples, Test = 25 K sample
 - Standardizing all features using StandardScaler (to scale and standardize).
 - Cross Validation mechanism to avoid overfitting (3 folds)
 - Random Forest (nestimators = 50, *max_features='auto'*, max_depth = 5)
 - Model Fit and Predict
 - Calculate/create metrics (Accuracy, Recall, Precision) and Confusion Matrix to analyze the performance of the model.

Model Performance

Confusion Matrix	Predicted NO	Predicted YES
Actual NO	TN	FP
Actual YES	FN	TP

Metric	Formula
Accuracy	$(TP + TN) / \text{Total}$
Recall (Sensitivity) % of wolves I actually identified	$TP / \text{Actual YES}$
Precision % of wolves when I cried 'Wolf'	$TP / \text{Predicted YES}$



Metric	Value
Accuracy	0.88
Recall	0.73
Precision	0.81

Result

model is using Random Forest.

- Accuracy is 88%
- OK for MVP, but, over all not a great process

Enhancements & Next Steps

- Enhance this model with new/more features to increase the accuracy (may be $\geq 90\%$)
- Adopting Principal Component Analysis (PCA) mechanism for better feature engineering (Aggregating the features).
- Need to analyze Precision/Recall graphs, ROC Curve and AUC for better interpretation and scale the performance.
- Other ensemble models
 - SVM (Support Vector Machine)
 - GBM (Gradient Boost Machine)
 - KNN
 - Deep Learning (no clue at this moment on how and what to do with this. Just an Idea as name and concept sounds Fancy (to me at least 😊))
 - Voting Classifier (hard/soft) to pick the right model.

Infrastructure

- Better Hardware
- Reduction in I/O times – SQL optimization

Who are intended Audience for this Model?

How will internal customers use the product

