



# Predicting Customer Churn Based on User Data

Providing Predictive Models and Data Insights to Aide in Digital  
Marketing and Improve Customer Retention

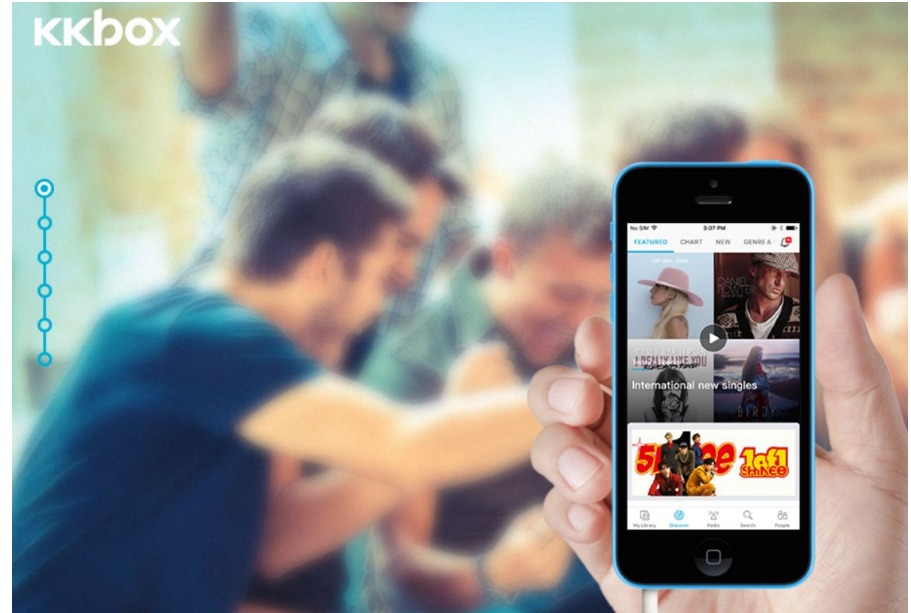


# Background: Business Problem

- For a subscription based business, customer churn refers to customers canceling their service
- The churn rate must be lower than the new subscriber rate for the company to grow
- Accurately predicting customer churn is critical for business projections and long term success
- Segmenting customers who have a high churn probability is very valuable to the marketing team to deploy targeted efforts to increase customer retention

# Hypothetical Client

- KKbox is the biggest music streaming service in East Asia with a large library of Asian pop music
- Users can stream unlimited numbers of songs from their smartphones or other devices in exchange for a monthly fee
- Basically, it's Taiwanese Spotify



# Data

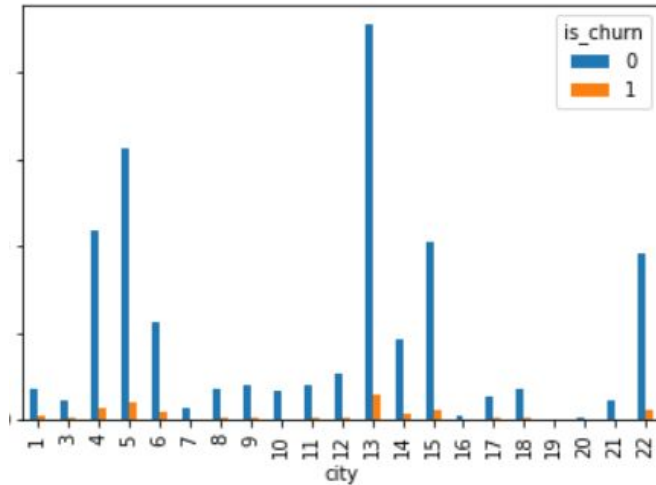
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000000 entries, 0 to 9999999
Data columns (total 9 columns):
msno          object
date          int64
num_25        int64
num_50        int64
num_75        int64
num_985       int64
num_100       int64
num_unq       int64
total_secs    float64
dtypes: float64(1), int64(7), object(1)
memory usage: 1.5 GB
```

- Kaggle provided their anonymized user data on the competition page.
- There were 4 different types of csv data tables: train/test, members, transactions, and user logs
- One column they all have in common is User ID (MSNO)
- Train/test and members files are indexed by User ID, while transactions and user logs are indexed by date
- Transactions file contain ~ 20 million records (@1.61 GB) and user logs file contains ~ 200 million (@ 22.4 GB) records
- This presents a challenge as most any laptop/desktop does not have enough RAM to hold these files in memory if using python packages such as Pandas taught in this course

# Data Description (columns)

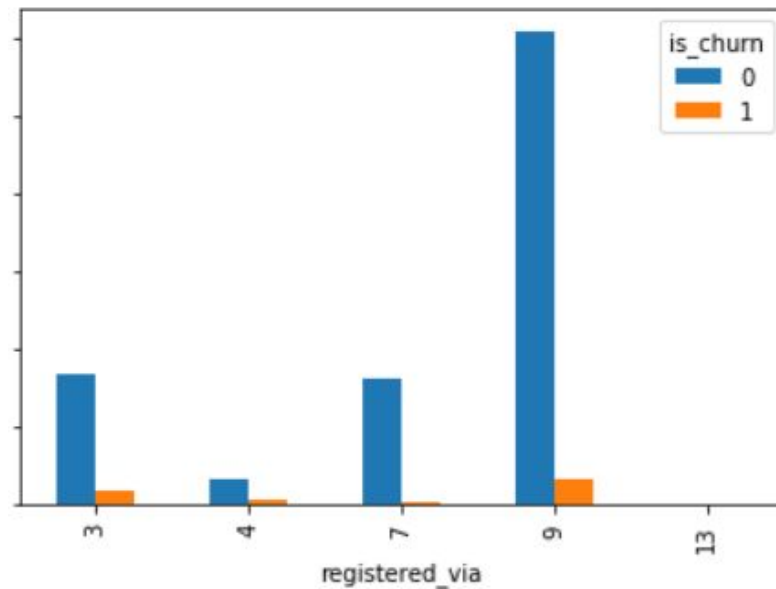
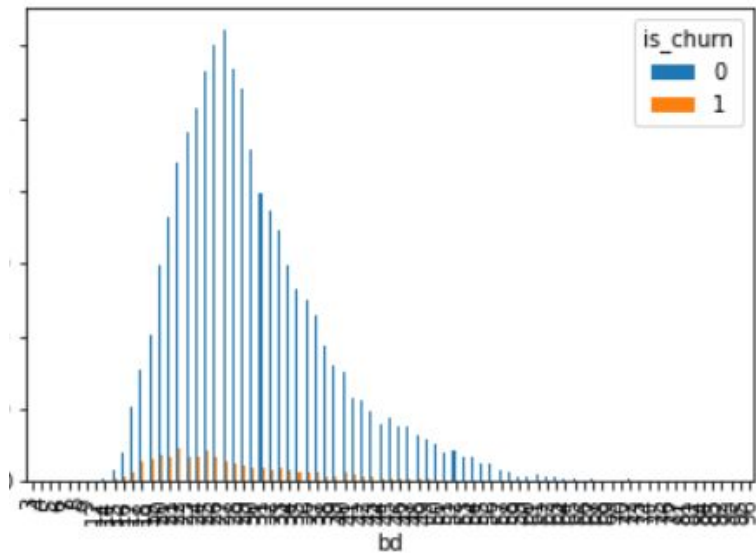
- Train/test
  - User ID
  - Churn (0 or 1)
- Members
  - User ID
  - Age
  - Gender
  - Registration method
  - Registration Date
- Transactions
  - Date
  - User ID
  - Payment method
  - Days in pay period
  - Payment plan list price
  - Actual amount payed
  - Auto renew (0 or 1)
  - Membership expire date
  - Cancellation (0 or 1)
- User Logs
  - Date
  - User ID
  - # of Songs played to 0-25%
  - # of Songs played to 25-50%
  - # of Songs played to 50-75%
  - # of Songs played to 75-98.5%
  - # of Songs played to 100%
  - # of unique songs played
  - Total seconds played

# Exploratory Data Analysis

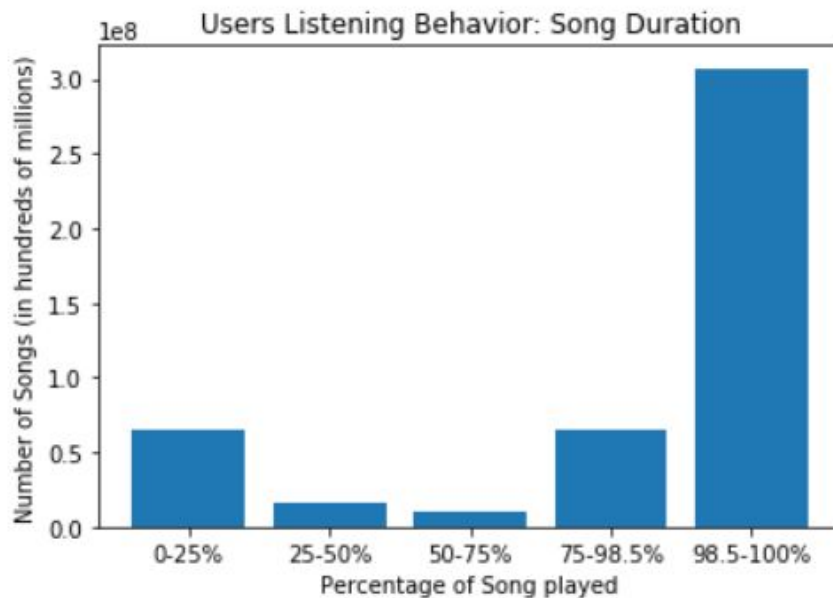
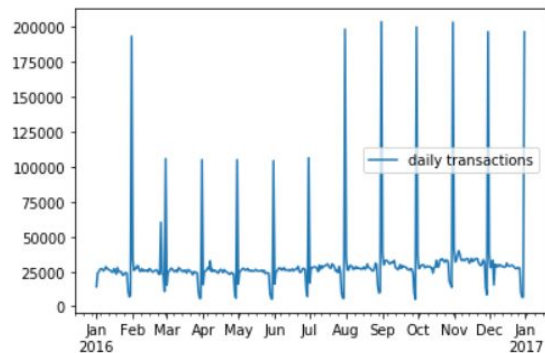
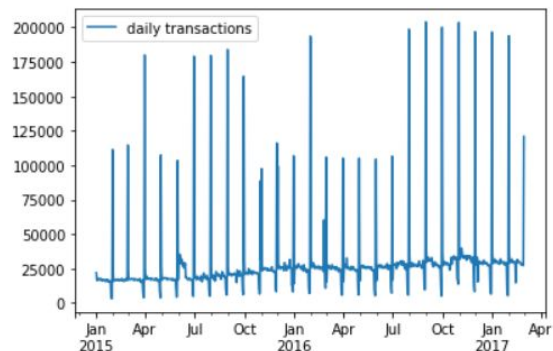


```
city
1    12.250000
3     7.203390
4     6.147186
5     5.772137
6     7.023411
7     3.875969
8     6.539510
9     5.450237
10    3.021148
11    5.825243
12    5.300353
13    5.732616
14    6.224490
15    5.050973
16    3.773585
17    5.535055
18    3.641457
19   28.571429
20    0.000000
21    4.803493
22    5.416051
Name: churn_percentage, dtype: float64
```

## EDA con'td

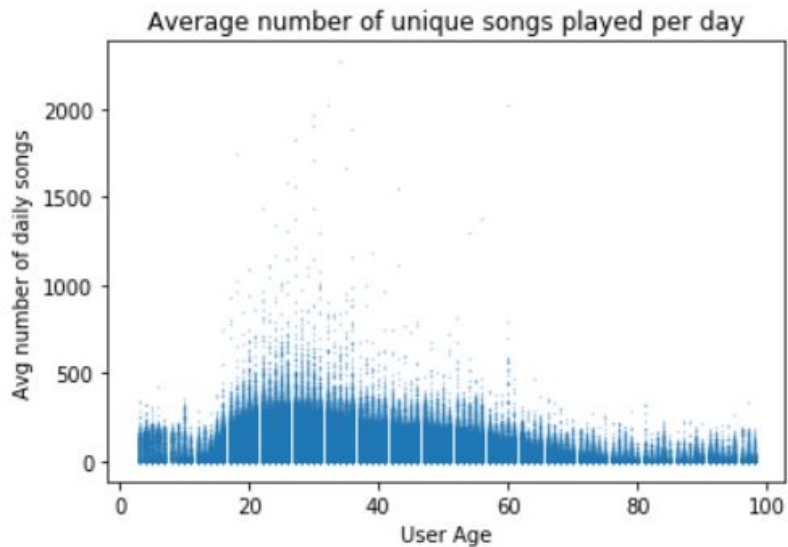
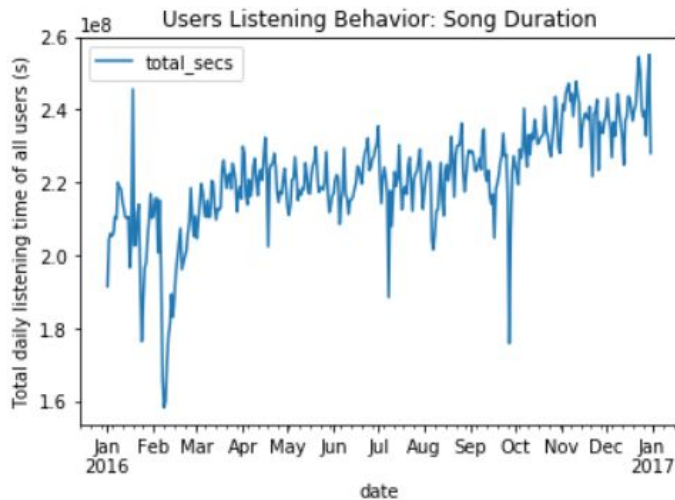


# EDA con'td





# EDA con'td



# Correlation Matrix (Pearson R)

	city	bd	registered_via	is_churn	num_25	num_50	num_75	num_985	num_100	num_unq	total_secs	days_logged
city	1.000000	-0.020445	0.045767	-0.022499	0.001224	0.001040	-0.006339	0.001040	-0.000279	-0.000679	-0.002455	0.003994
bd	-0.020445	1.000000	0.209202	-0.071810	-0.089061	-0.090175	-0.086469	-0.090175	0.022526	-0.040294	0.010269	-0.035512
registered_via	0.045767	0.209202	1.000000	-0.073980	-0.031941	-0.031360	-0.021466	-0.031360	0.010942	-0.007058	0.007218	-0.019007
is_churn	-0.022499	-0.071810	-0.073980	1.000000	0.005029	0.003880	0.013094	0.003880	-0.022791	-0.013198	-0.022951	-0.046399
num_25	0.001224	-0.089061	-0.031941	0.005029	1.000000	0.533542	0.422364	0.533542	0.069331	0.414502	0.118600	0.021858
num_50	0.001040	-0.090175	-0.031360	0.003880	0.533542	1.000000	0.591780	1.000000	0.078209	0.325399	0.143024	0.032062
num_75	-0.006339	-0.086469	-0.021466	0.013094	0.422364	0.591780	1.000000	0.591780	0.102863	0.301384	0.173186	0.031214
num_985	0.001040	-0.090175	-0.031360	0.003880	0.533542	1.000000	0.591780	1.000000	0.078209	0.325399	0.143024	0.032062
num_100	-0.000279	0.022526	0.010942	-0.022791	0.069331	0.078209	0.102863	0.078209	1.000000	0.783419	0.982416	0.076924
num_unq	-0.000679	-0.040294	-0.007058	-0.013198	0.414502	0.325399	0.301384	0.325399	0.783419	1.000000	0.819015	0.068628
total_secs	-0.002455	0.010269	0.007218	-0.022951	0.118600	0.143024	0.173186	0.143024	0.982416	0.819015	1.000000	0.080310
days_logged	0.003994	-0.035512	-0.019007	-0.046399	0.021858	0.032062	0.031214	0.032062	0.076924	0.068628	0.080310	1.000000

# Data Wrangling

Out[12]:

	msno	is_churn	bd	registration_init_time	num_25	num_50	num_75	num_95	num_100	num_unq	...	city_22.0	registered_via_0.0
3ZaolqOUAZPsH1q0teWCg=	1	28.0	20131223.0	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0	0
Udr+E+3+oewwweYz9cCQE=	1	20.0	20131223.0	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0	0
hiRnAVgibMyazbCxvWPCg=	1	18.0	20131227.0	5.166667	1.833333	1.500000	1.833333	8.000000	15.000000	...	...	0	0
rGCG2Ecrogbc2Vy5YhsfhQ=	1	0.0	20140109.0	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0	0
lw/XKpMgrEMdG2edFOxA=	1	35.0	20140125.0	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0	0
D99M5vYB3CN2HzkEY+eM=	1	0.0	20140126.0	1.500000	0.000000	0.000000	0.000000	2.750000	5.000000	...	...	1	0
lJfFoxD1EcKYCc76F5IAWw=	1	0.0	20140129.0	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0	0
4dXMLk0jOn65d7a8tQ2Eds=	1	28.0	20140202.0	0.500000	0.166667	0.166667	0.166667	33.500000	17.666667	...	...	0	0
JbszOMXw3kay/1AIZCq3Ebl=	1	21.0	20140212.0	14.857143	2.285714	1.571429	2.285714	46.285714	63.142857	...	...	0	0
eN3oaNmhdmtKooF2iRYEE=	1	0.0	20140228.0	47.500000	2.833333	0.166667	2.833333	21.666667	61.666667	...	...	0	0
Cezv5KBK7+DlMujNlbYgylrs=	1	0.0	20140307.0	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0	0
<1Bc6g+7LFKzoNf+zlJtDoQ=	1	0.0	20140323.0	1.500000	1.000000	1.500000	1.000000	16.833333	21.333333	...	...	0	0
FerqTEgmno3x7Rc7YGwzw=	1	32.0	20140324.0	3.000000	1.285714	0.857143	1.285714	10.285714	14.285714	...	...	0	0
CVYPdek7K4Leu+aqbCRo8=	1	0.0	20140402.0	5.000000	2.000000	0.500000	2.000000	40.000000	32.125000	...	...	0	0
zB9gChiSR4tWP4lvJGdxSM=	1	21.0	20140407.0	3.000000	1.600000	1.800000	1.600000	90.000000	91.200000	...	...	0	0
siKwhz0Za0yU2GIQtKa1JFk=	1	20.0	20140415.0	14.666667	4.000000	1.500000	4.000000	2.500000	21.666667	...	...	0	0
fSxP0VrzDsIB0Hdth12gqQE=	1	0.0	20140420.0	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0	0
ucgyuT2i8JliJ87bOlSxtFvBw=	1	0.0	20140425.0	15.000000	2.500000	2.500000	2.500000	27.000000	44.000000	...	...	0	0
VnjzY7U1G24mVFNdzGNQ=	1	29.0	20140510.0	6.571429	1.285714	1.000000	1.285714	21.142857	27.857143	...	...	0	0
yfme2pHLE2y+RJ3eGcLT0k=	1	0.0	20140515.0	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0	0
DnSPi16Y2x5C+uVFEhY-	1	0.0	20140605.0	2.333333	1.000000	0.500000	1.000000	14.166667	17.833333	...	...	0	0

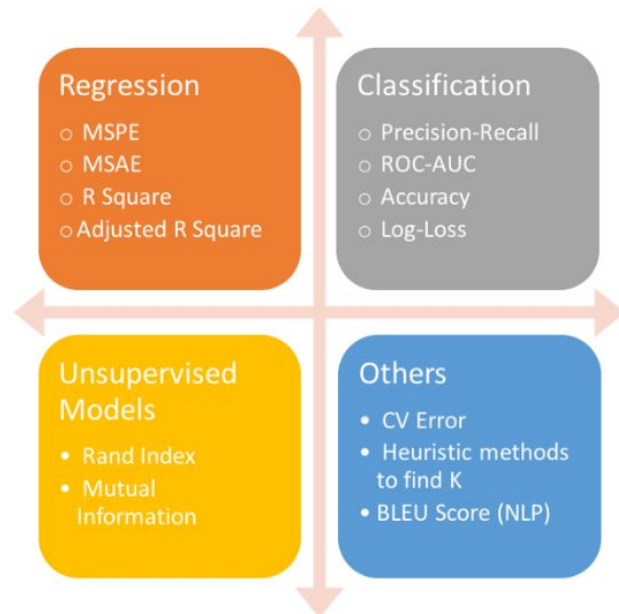
# Model Selection

The evaluation metric for this competition is **Log Loss**

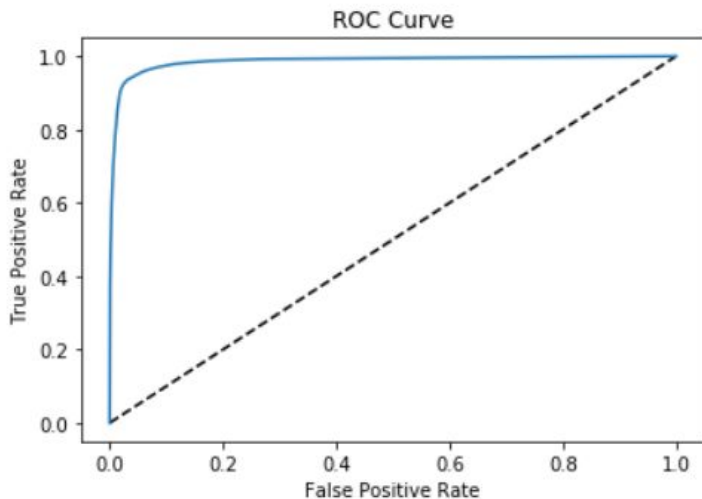
$$\text{logloss} = -\frac{1}{N} \sum_{i=1}^N (y_i \log(p_i) + (1 - y_i) \log(1 - p_i))$$

Log loss of random guessing =  $\ln(0.5) = 0.693$

Log loss of hypothetical perfect model = 0.0



# Model Performance: Random Forest



AUC: 0.9861211554440634

Confusion  
Matrix

```
[[174343  2371]
 [   2621 14857]]
```

Evaluation  
Set:

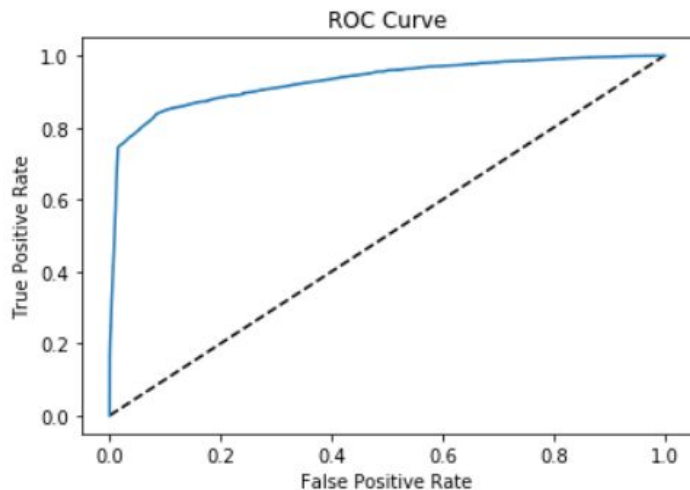
```
#check log loss
log_loss(y_test,y_pred_prob)
```

0.09625370790222848

Competition  
Scoring:

0.30744

# Model Performance: Logistic Regression with Grid Search Hyperparameter Optimization



AUC: 0.930369867332425

Confusion  
Matrix

```
[[174087  2627]  
 [  4606 12872]]
```

Evaluation  
Set:

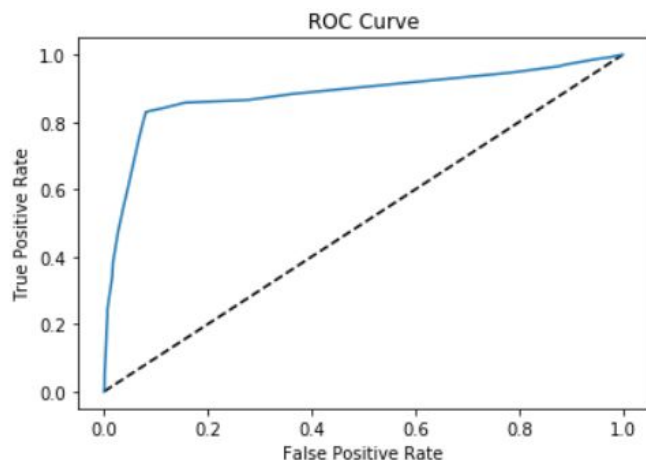
```
#check log loss  
log_loss(y_test,y_pred_prob)
```

0.13416865382929385

Competition  
Scoring:

0.26963

# Model Performance: Logistic Regression- ridge regression penalty with LASSO regression selected features (final model)



AUC: 0.8817120310353784

Confusion  
Matrix

```
[[173671  3043]
 [ 10788  6690]]
```

Evaluation  
Set:

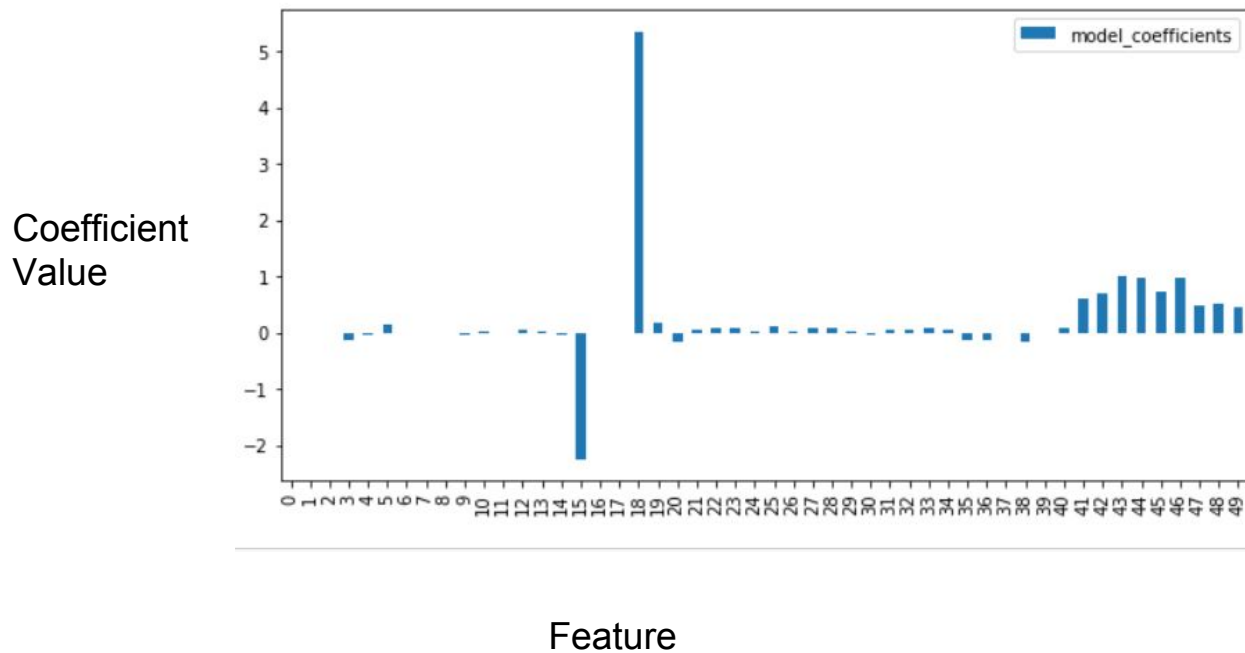
```
#check log loss
log_loss(y_test,y_pred_prob)
```

0.1738358467535548

Competition  
Scoring:

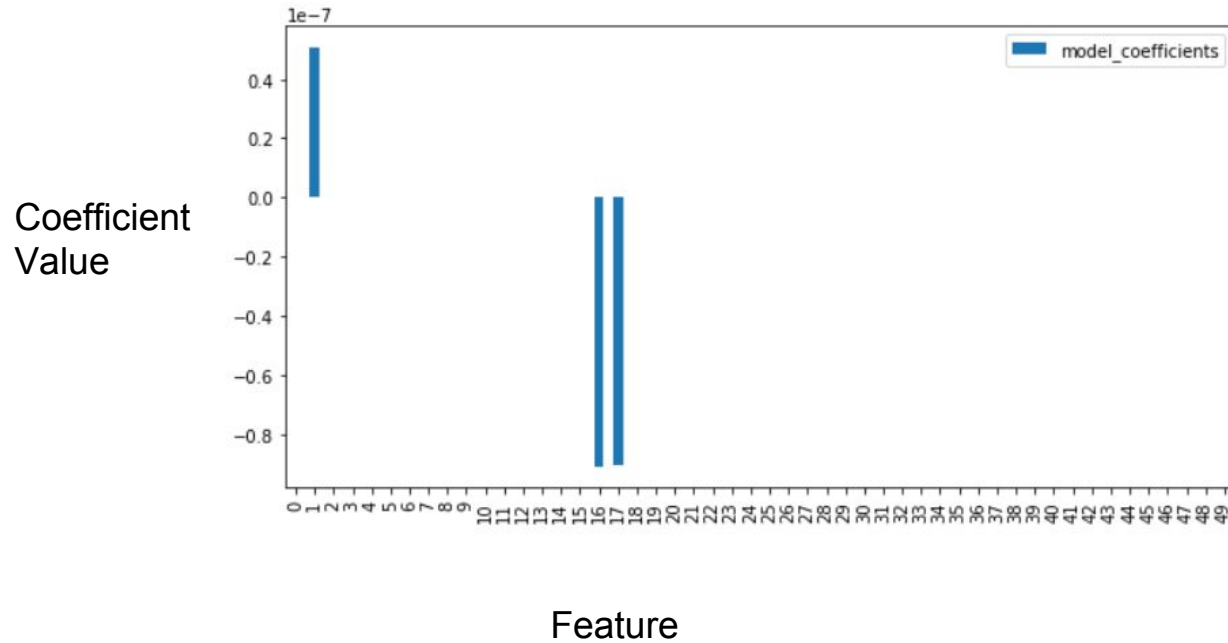
**0.13673**

# Features: LASSO Regression with All Features

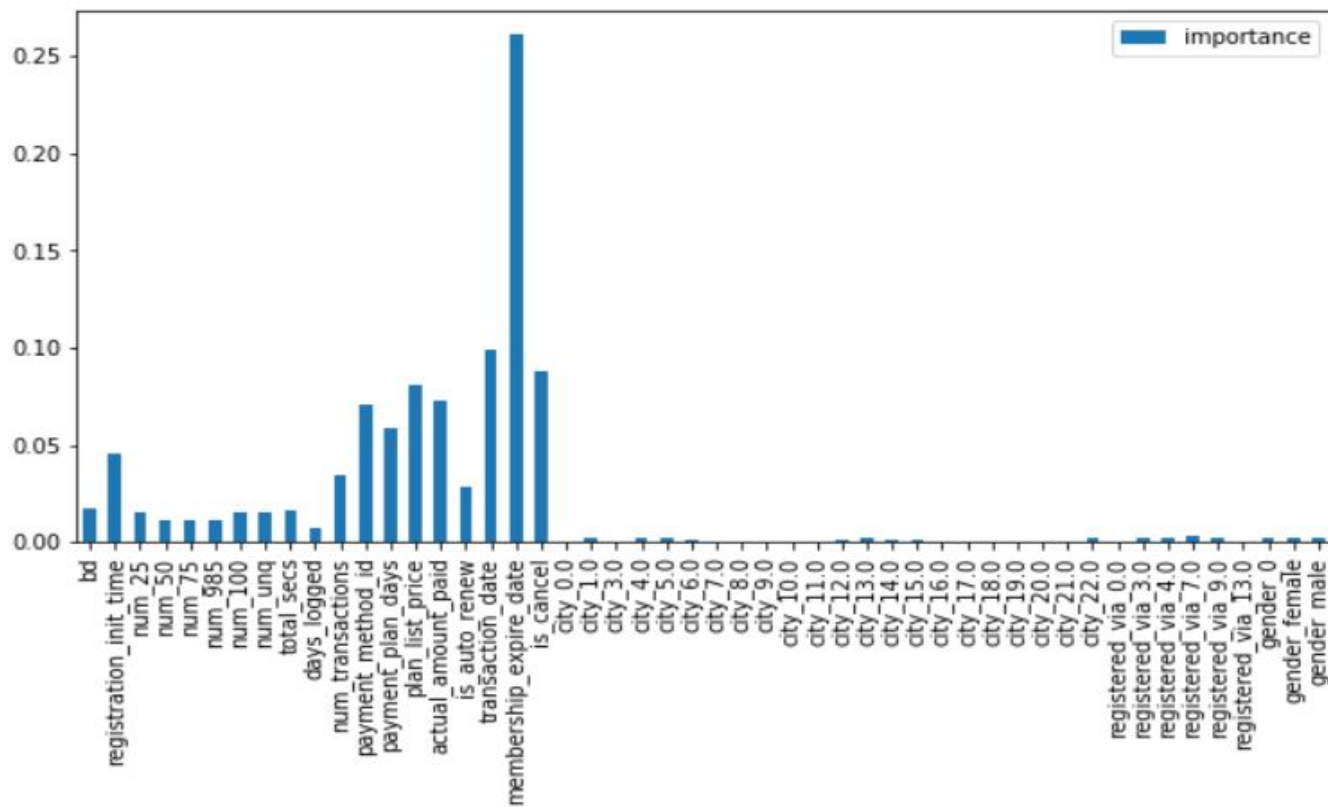




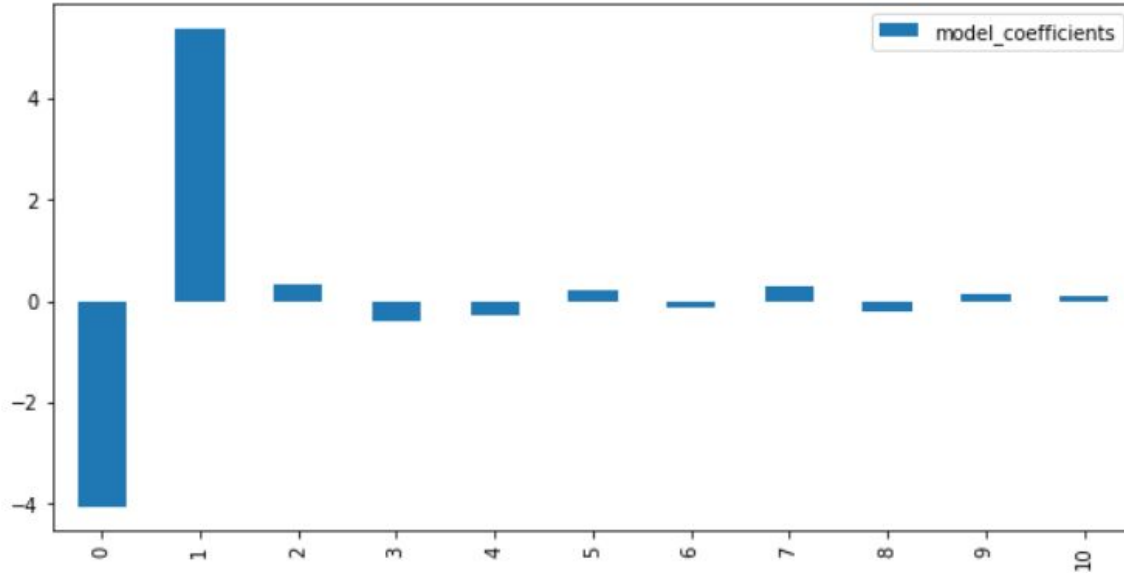
# Features: Ridge Regression with All Features



# Random Forest Feature Importance



# Final Model Feature Coefficients



	features	model_coefficients
0	is_auto_renew	-4.052154
1	is_cancel	5.375505
2	registered_via_0.0	0.317831
3	registered_via_3.0	-0.394750
4	registered_via_4.0	-0.274589
5	registered_via_7.0	0.210269
6	registered_via_9.0	-0.119956
7	registered_via_13.0	0.286762
8	gender_0	-0.219702
9	gender_female	0.151013
10	gender_male	0.094256

## Recommendation to Client:

- Apply targeting marketing strategies to customers with high churn probability
- Consider features of users that have significance to churning when implementing product improvements