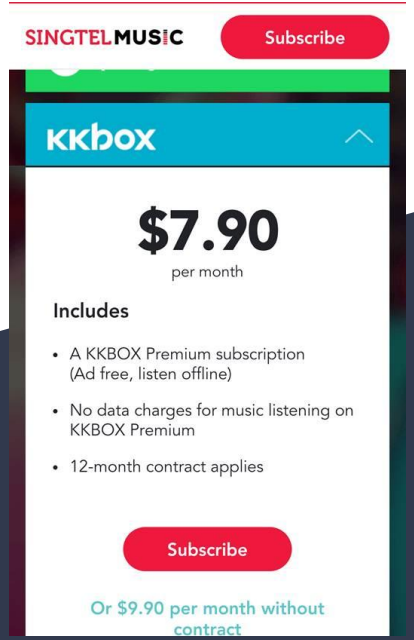
Dealing with Imbalanced Data Set- KKBox Churn Rate Prediction

Xiatong Gui PO'19
Minh-Quan Do PO'19
Zihao Xu PO'19



Motivation

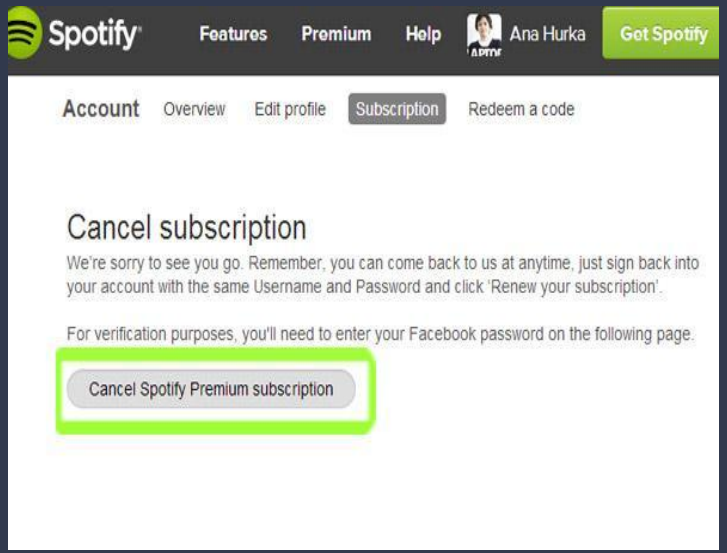
What are we doing?

- KKbox is a music streaming platform in Asia (Like Spotify!)
- Building a model to predict whether a user will churn when their subscription to KKBox expires

Why are we doing this?

- Important for marketing strategy
- To explore complicated models that improves prediction accuracy
- To practice handling “unwieldy” data
- Eyes on the prize ($$$)

Understanding “Churn”



What is “churn”?

TL;DR: When users end their subscription to your (company's’) service!

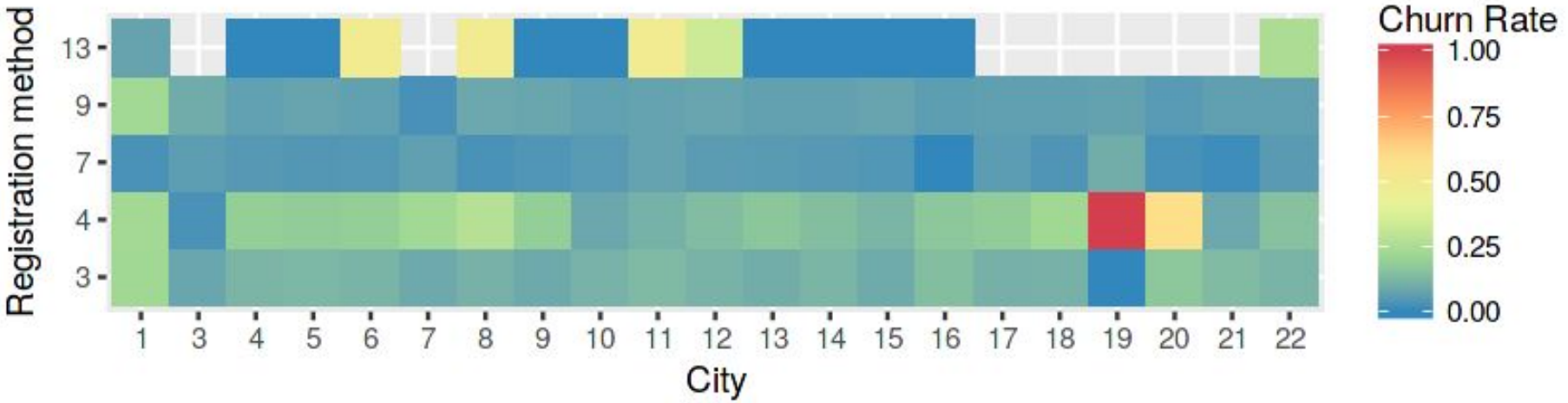
More details

- KKBox’s subscription model: 30 days -> users resubscribe every month
- criteria: no new valid service subscription within 30 days after the current membership expires

A Brief Summary of Our Data

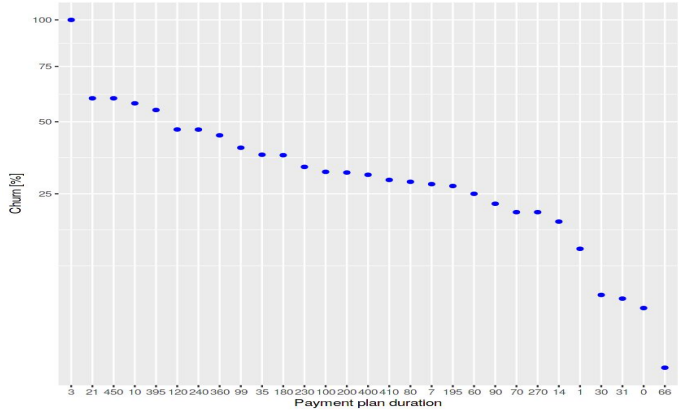
Dataset	Variables
Training Set	user_id, **is_churn**
Member Portfolio	user_id, city, age, gender registered_method, registration_init_time
Transaction Information	user_id, payment method, payment plan, plan price, actual amount paid, auto renew, transaction date, membership expire date, cancel
Login Information	user_id, # of songs played less than 25/50/75/98.5/100% of song length, # of unique songs played, total seconds played
Test Set	user_id, **is_churn**

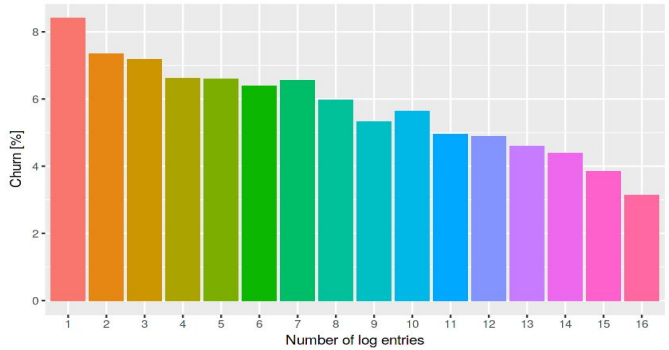
Relationship Between Predictors and Churn Rate

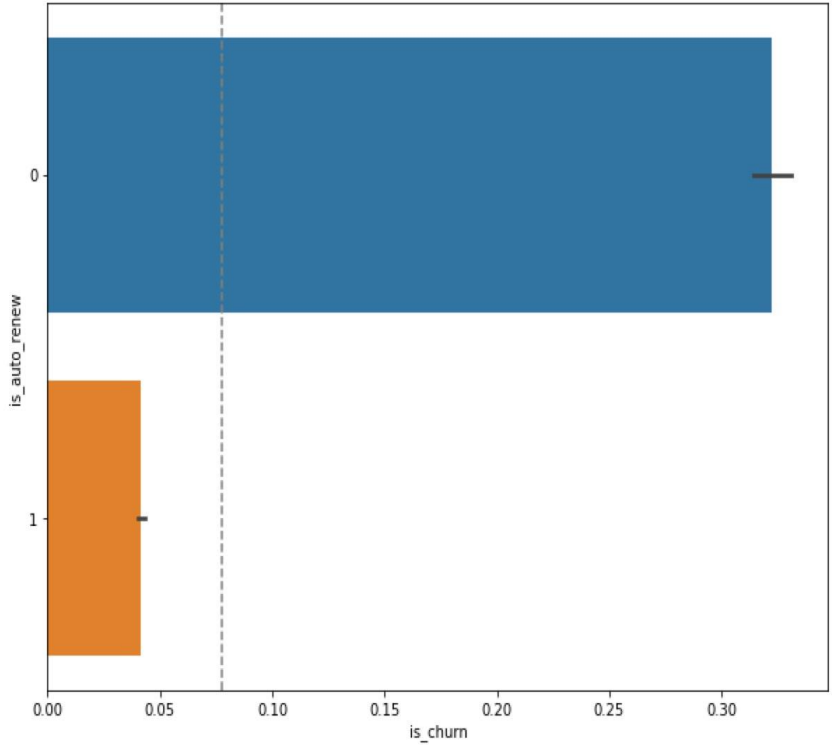


Transaction and User Log and Churn Rate

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







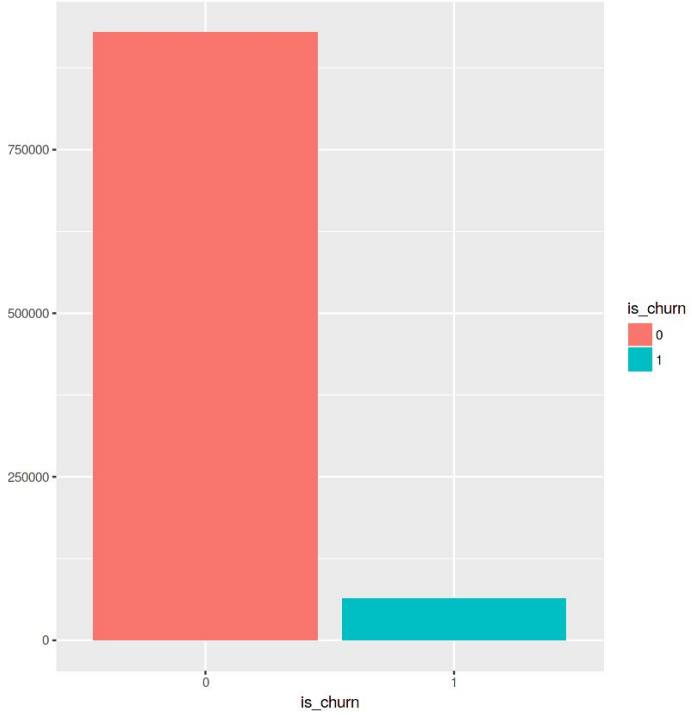
Technical Challenges

Very Large Datasets:

- user_log.csv has 30 GB, 2 billion entries
- **Use Kaggle kernel!**

Imbalanced response variable:
churn/no churn rate is 1:15.5

- Accuracy is unreliable (predicting all no churn yields 94% accuracy)



Technical Challenges

Messy Features:

- Each user has multiple transaction and log in data
  - Furthermore, users who don’t churn have more transaction and log in -> **exagerate no churn when merge**

Outliers

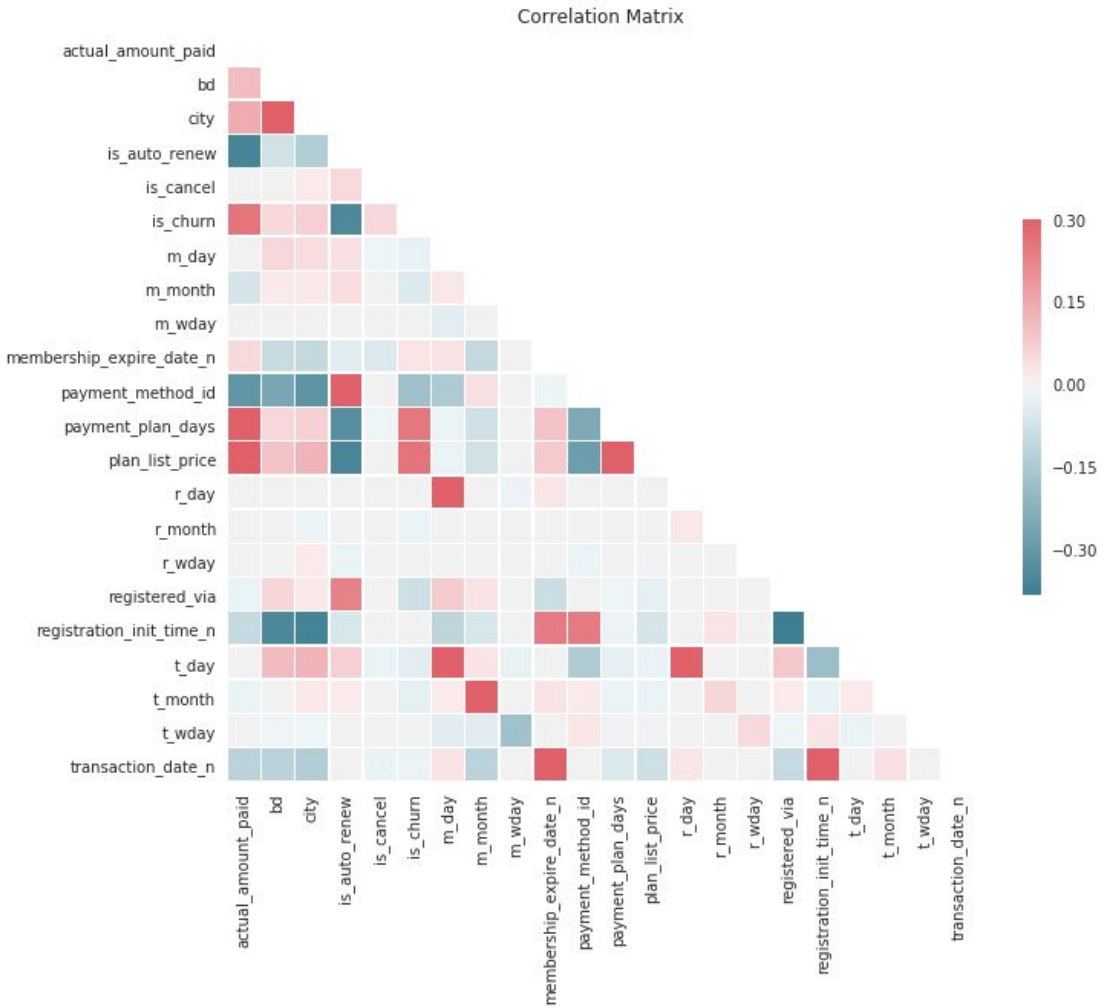
- Numerous features have outliers:
  - Birthday -> 1000 yrs old user
  - Total secs: negative value
  - Date: beyond the scope
  - Gender: missing info

Focus 1

Feature Selection

Feature Selection:

Over 100 features
--------->
Fun fact:
47 features included!



Feature Engineering:

1. Time:
   day, weekday, month
2. Hidden info:
   # trans, # logs
3. Dummy encoding:
   a. Correlation Matrix
   b. Counts
   c. Average churn rate

 # Features in the original data set:
['is_churn', 'bd', 'registration_init_time', 'actual_amount_paid', 'is_auto_renew', 'transaction_date', 'membership_expire_date', 'is_cancel', 'num_100', 'num_25', 'num_unq', 'total_secs']

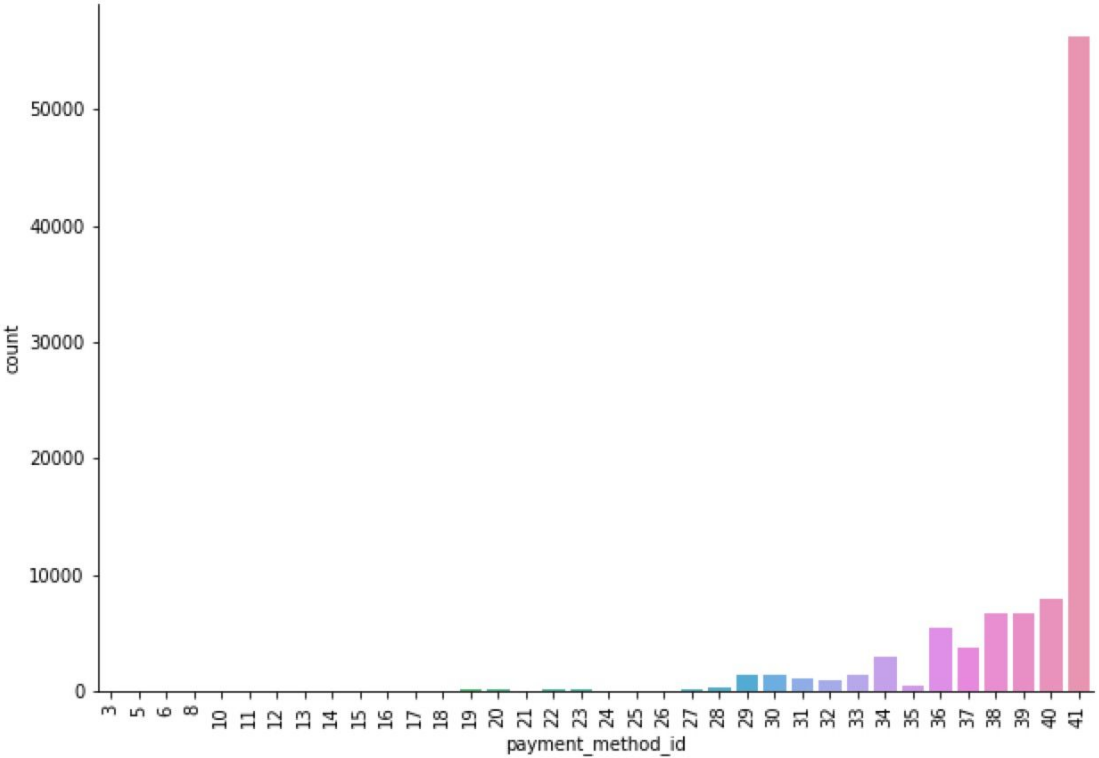
 # Time features:
['last_user_log_date', 't_month', 't_day', 't_wday', 'm_month', 'm_day', 'm_wday', 'r_month', 'r_day', 'r_wday', 'l_day', 'l_wday', ...]

 # Hidden info:
['trans_count', 'logs_count']

 # Dummy encodings:
['pm_id_41', 'pm_id_40', 'pm_id_39', 'pm_id_38', 'pp_days_30', 'pp_days_0', 'pp_days_31', 'city_11', ...]

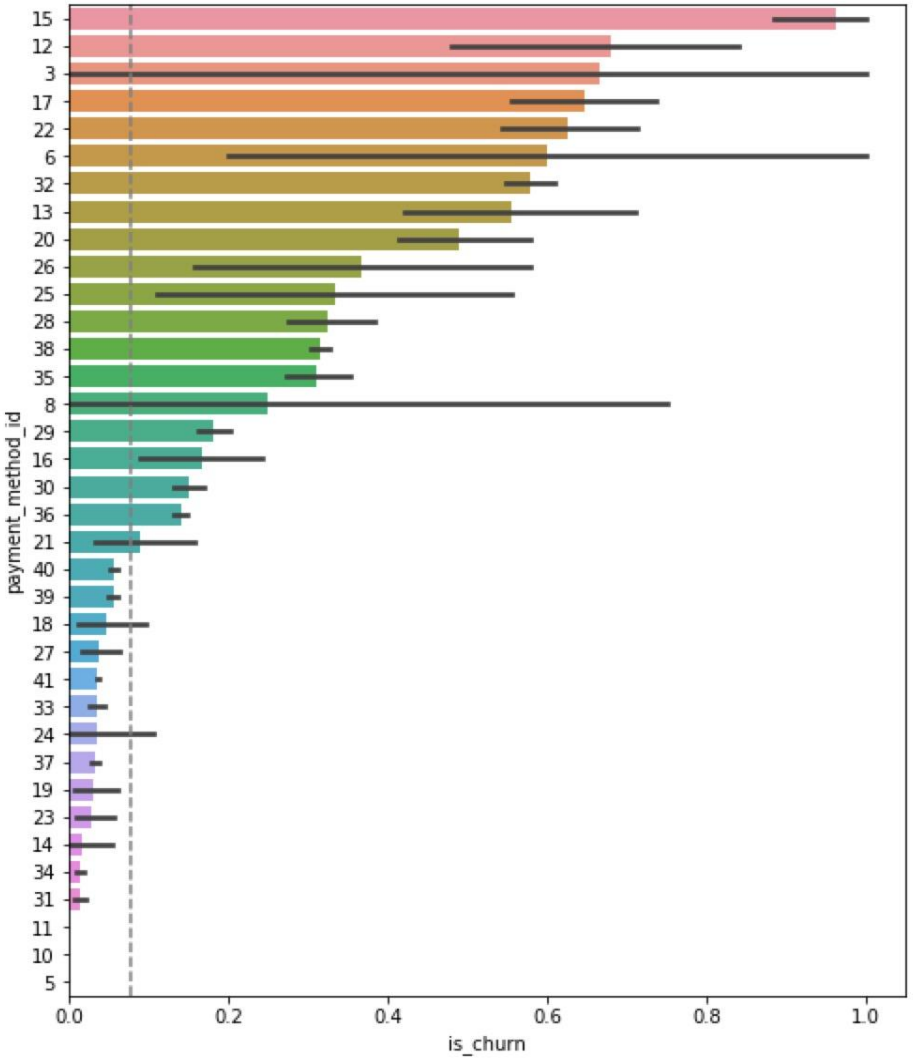
Dummy encoding:

Payment Method id



Dummy encoding:

Payment Method and Churn Rate



Focus 2

Tackle Imbalanced Data in Machine Learning

New Methods:

- Over-sampled in Random Forest (SMOTE package)
- Autoencoder (Neural Net)

New Performance Metrics:

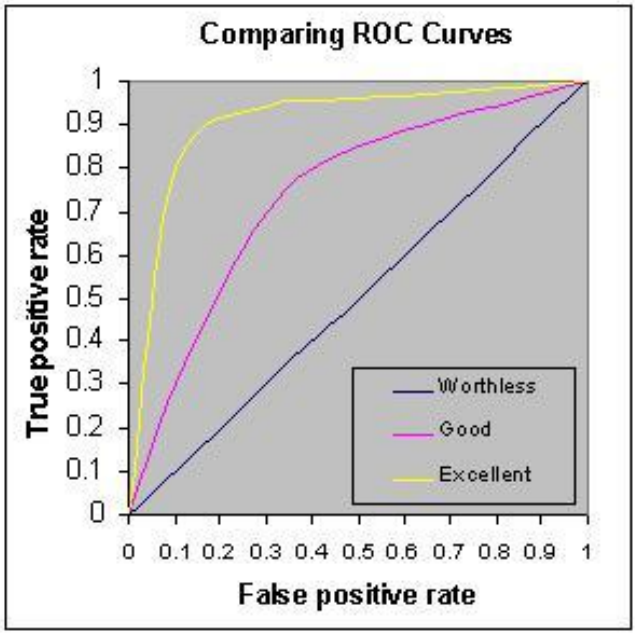
- Area under ROC Curve
- Precision-Recall Plot (Optional)

ROC Curve

ROC Curve: Receiver Operating Characteristic

Area Under the Curve (AUC)

- visualize **all** possible classification thresholds
- is a useful metric for datasets with highly unbalanced classes

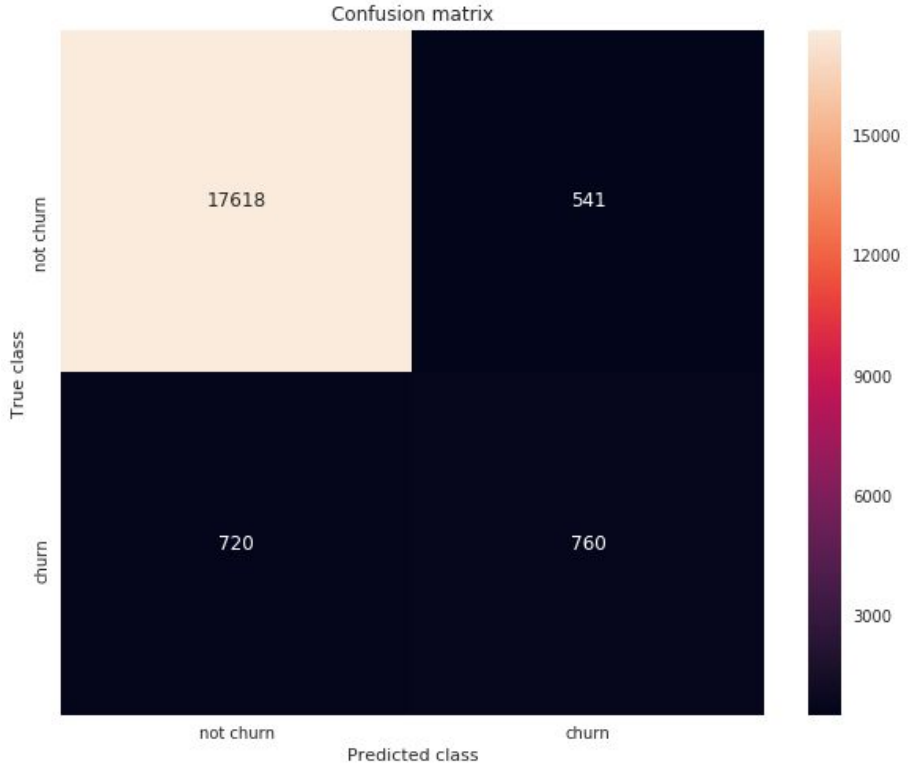


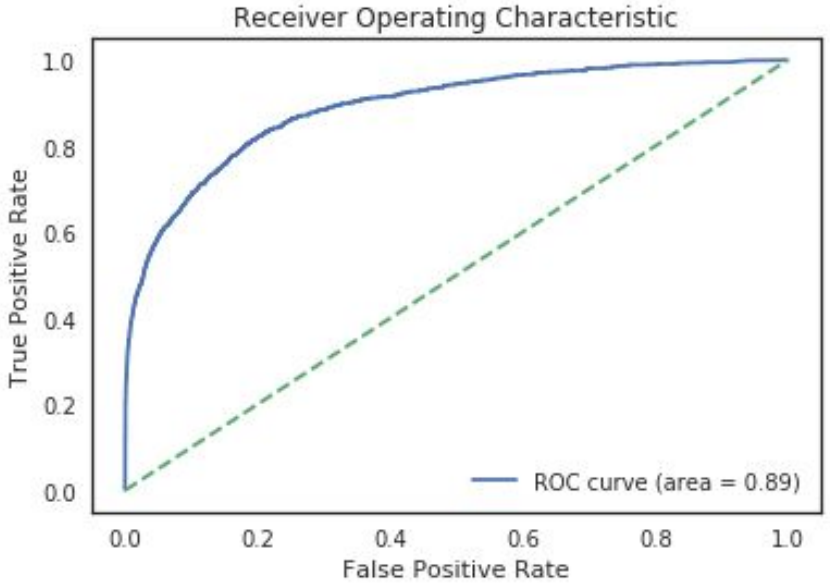
Model 1: Oversampled Random Forest

Algorithm:

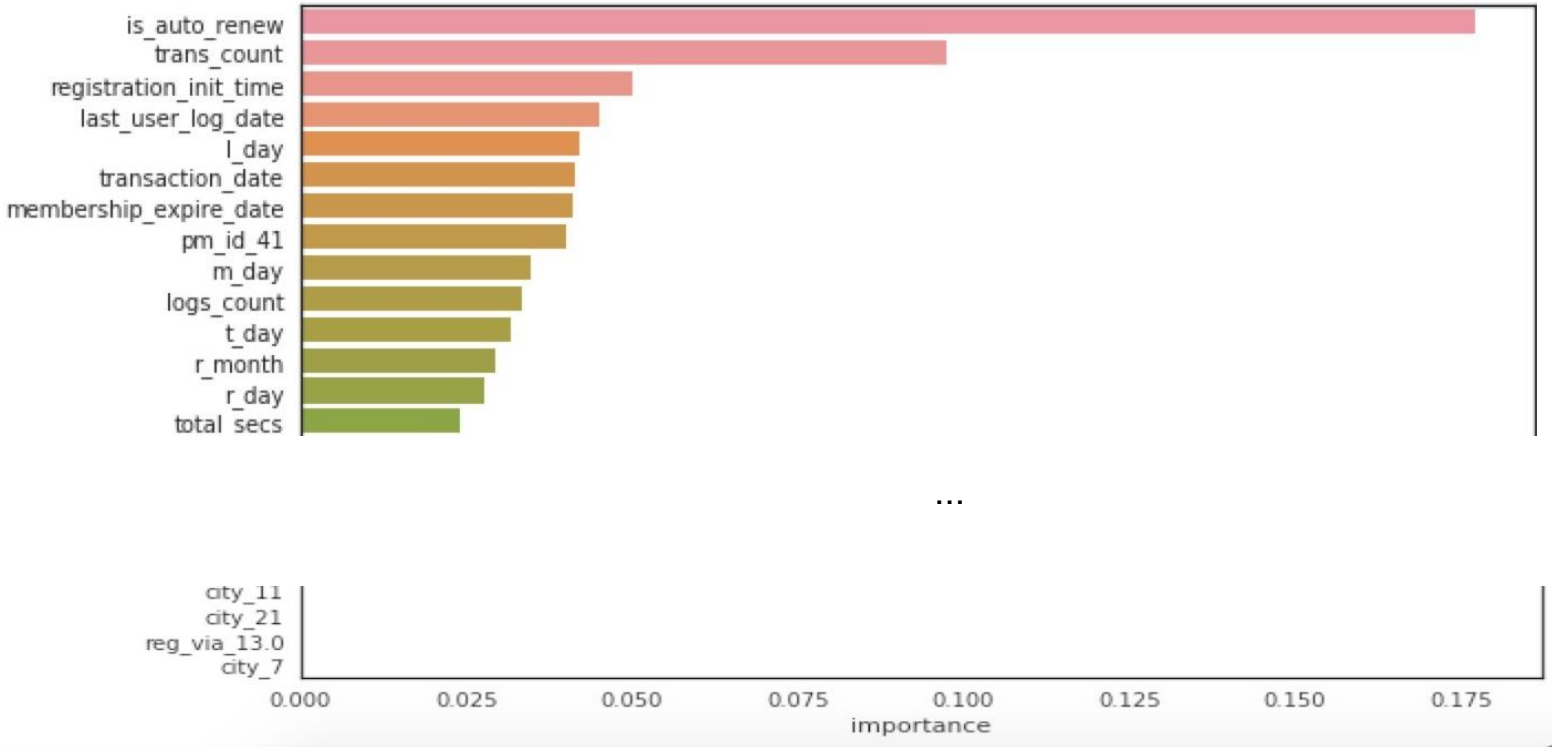
- (Optional) Preprocess the features
- Split to Train / Test sets
- **Oversample the obs. from churn class during bootstrap**
- CV using OOB to tune “mtry”, “min_samples_leaf (max-depth)”
- Model Evaluation: confusion matrix and Area Under the Curve (AUC)

Results


Confusion matrix
17618
541
720
760
not churn
churn
True class
not churn
churn
Predicted class
15000
12000
9000
6000
3000

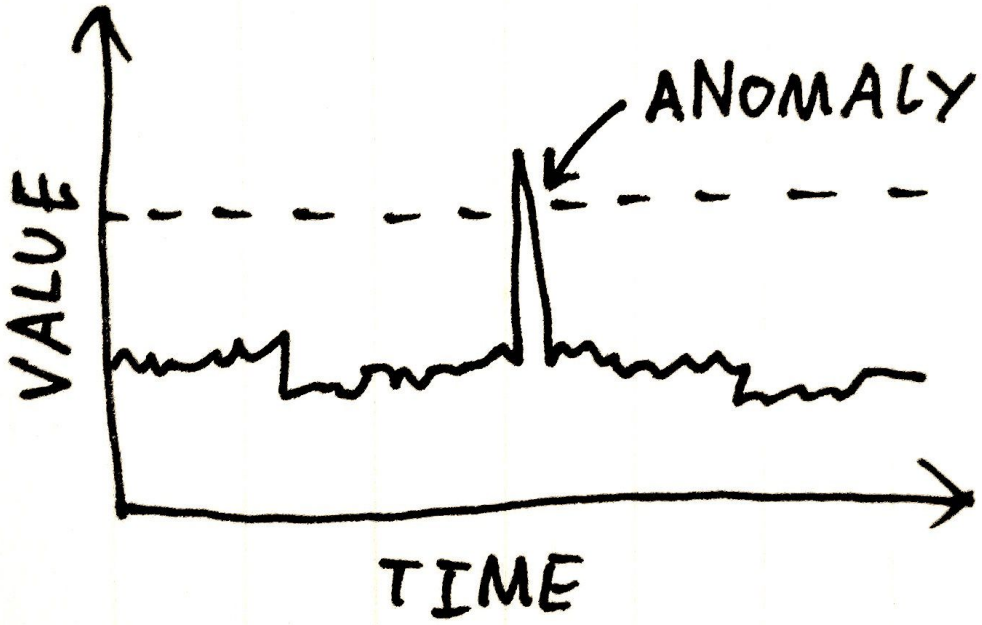

Receiver Operating Characteristic
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
ROC curve (area = 0.89)

Variable Importance



Model 2: Autoencoder

- Anomaly Detection and Neural Nets
- Particularly useful in dealing with imbalanced data set
- Applications: fraud detection



Reference: http://amid.fish/images/simple_anomaly.jpg

Anomaly Detection

Model Training:

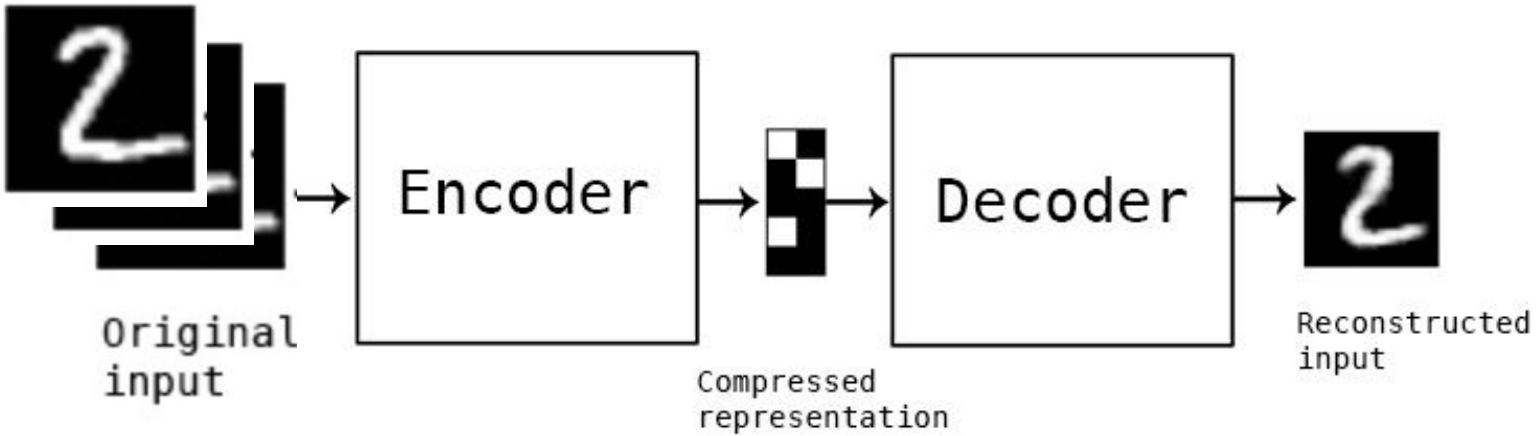
- Train the model only on the **normal**, “no_churn”, obs.
- Find the **patterns** of the normal group

Prediction:

- Feed new observations to the trained model
- The ones that **deviates** the most are predicted as “anomaly” or “churn”

Model 2: Autoencoder

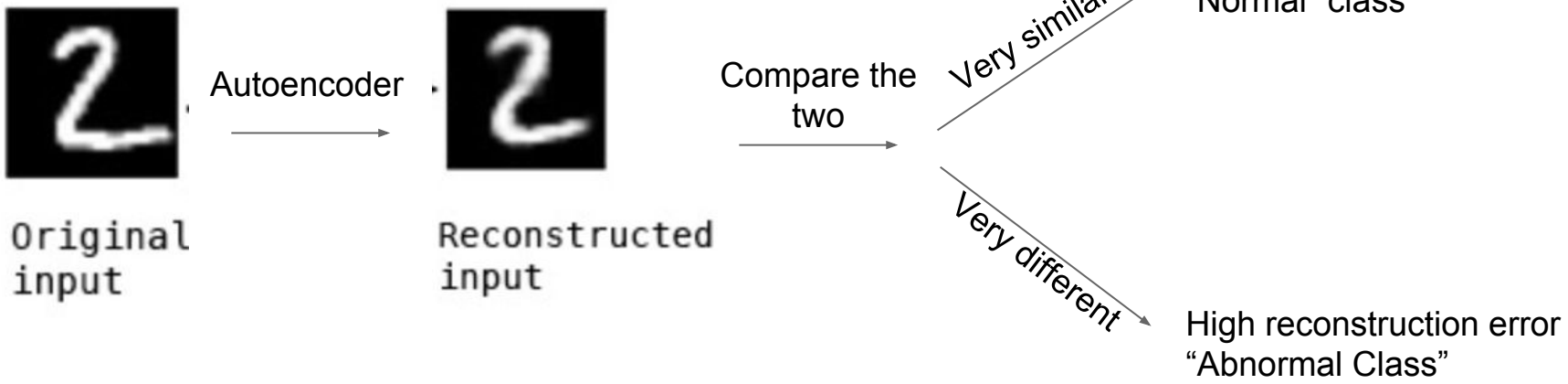
Encode: lower dimensional representation of features
Decode: reconstruct the original features from the Compressed Data



Reference: https://blog.keras.io/img/ae/autoencoder_schema.jpg

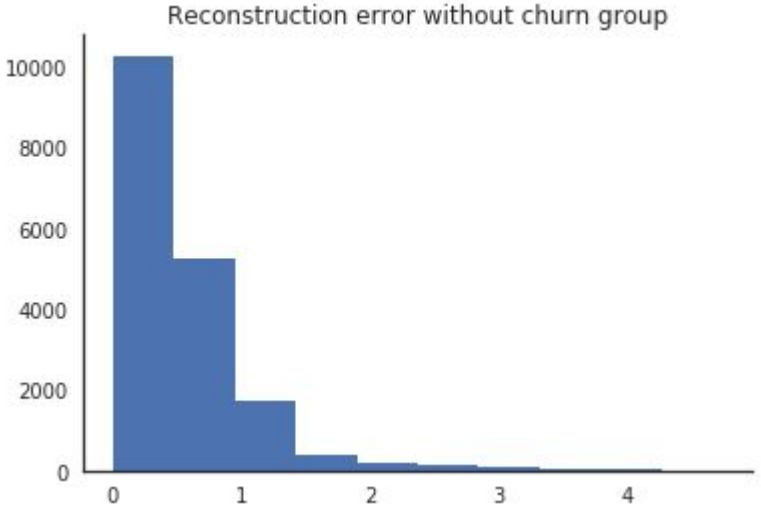
Model 2: Autoencoder

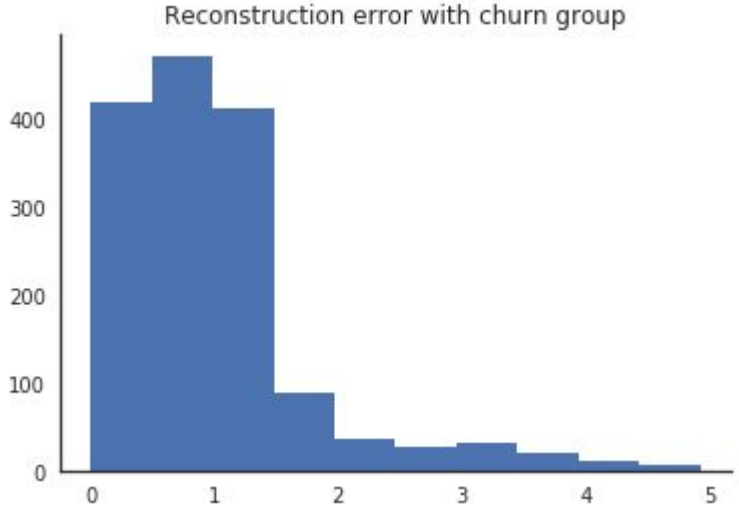
Reconstruction Error:
A numeric measure of how **different** are the reconstructed inputs from the original inputs



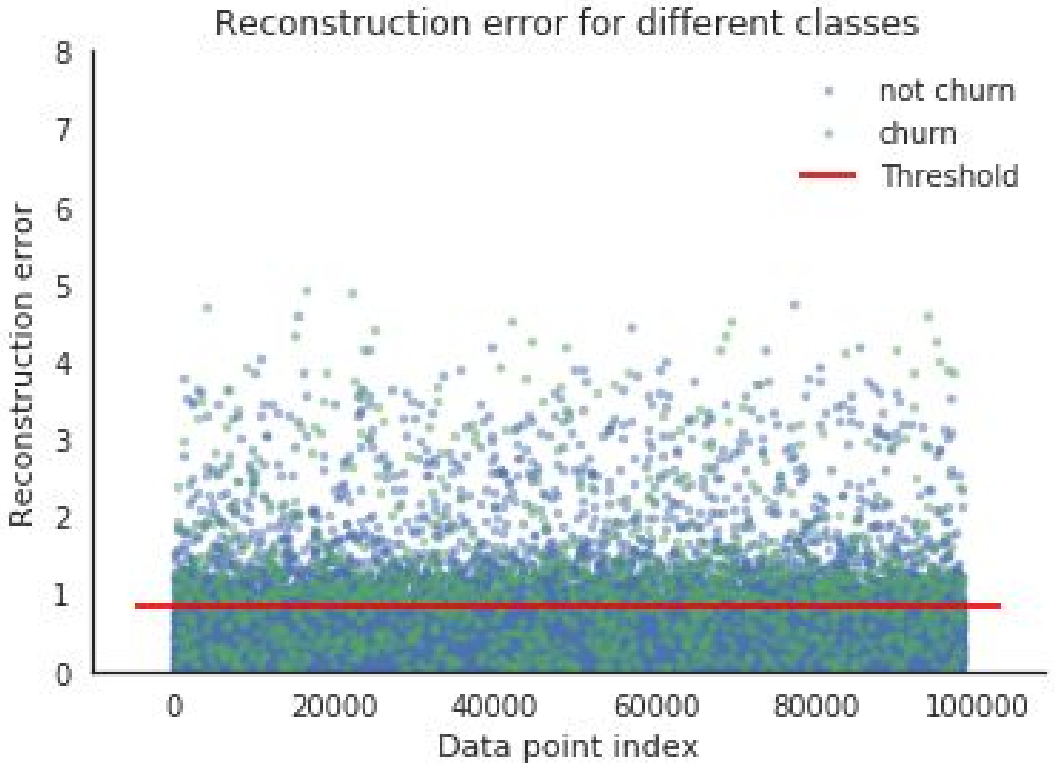
Training and Reconstruct

Train: using 73,842 **no_churn** observations
Test: using 19,639 **mixed** observations

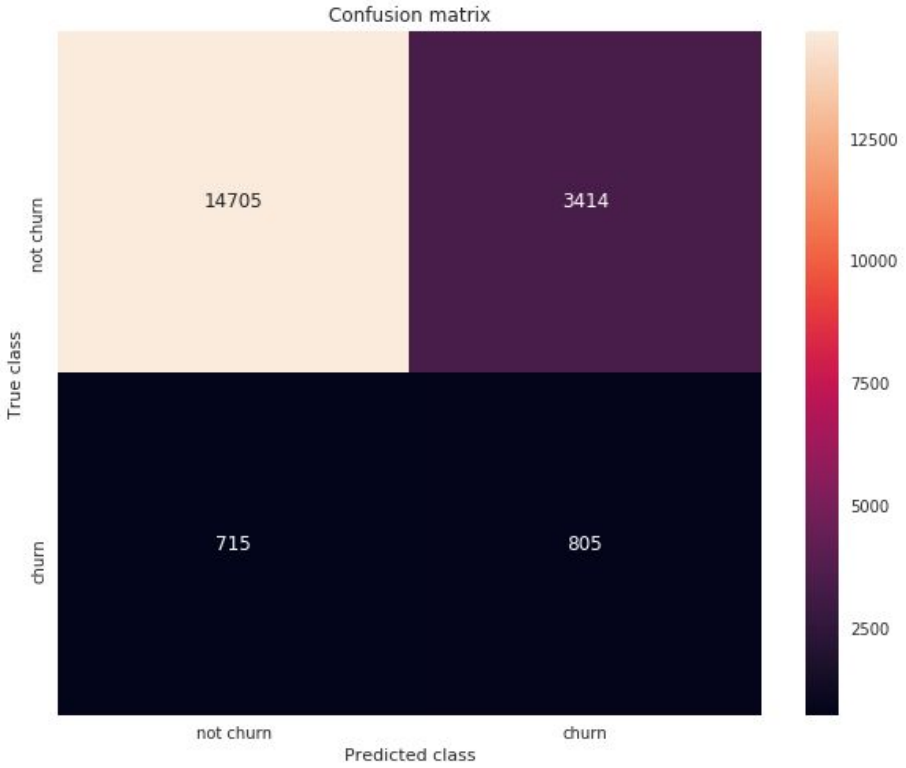


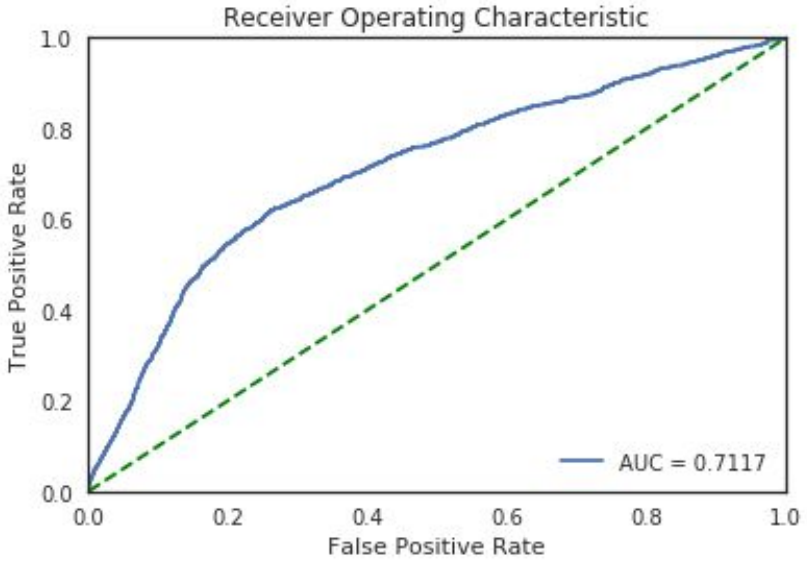


Reconstruction error w/ churn group



Results


Confusion matrix
14705
3414
715
805
not churn
churn
True class
not churn
churn
Predicted class
12500
10000
7500
5000
2500


Receiver Operating Characteristic
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
AUC = 0.7117

Results

Thoughts:

- Not as good as Oversampled-RF solely based on the AUC measure
- But! We can adjust the **cut_off** for reconstruction error!

Summary

What we've talked about:

- Challenges of dataset:
  - Big and messy data
  - Imbalanced class

Key Takeaway:

Feature Selection:

- Identify key features
- Dummy encoding

Tackle Imbalanced Data:

- New Performance Metrics
  - ROC Curve
- New Prediction Models
  - Oversampled RF
  - Autoencoder

Future Potential

Next steps:

- Try more advanced models:
  - XGBoosting
  - Other forms of Neural Nets

Bibliography

1. Yaya Xie, Xiu Li, E.W.T. Ngai, Weiyun Ying, Customer churn prediction using improved balanced random forests, In Expert Systems with Applications, Volume 36, Issue 3, Part 1, 2009, Pages 5445-5449, ISSN 0957-4174, https://doi.org/10.1016/j.eswa.2008.06.121
2. https://shiring.github.io/machine_learning/2017/05/01/fraud
3. https://medium.com/@curiousily/credit-card-fraud-detection-using-autoencoders-in-keras-tensorflow-for-hackers-part-vii-20e0c85301bd
4. https://lazyprogrammer.me/a-tutorial-on-autoencoders/