Clicks Activation Study

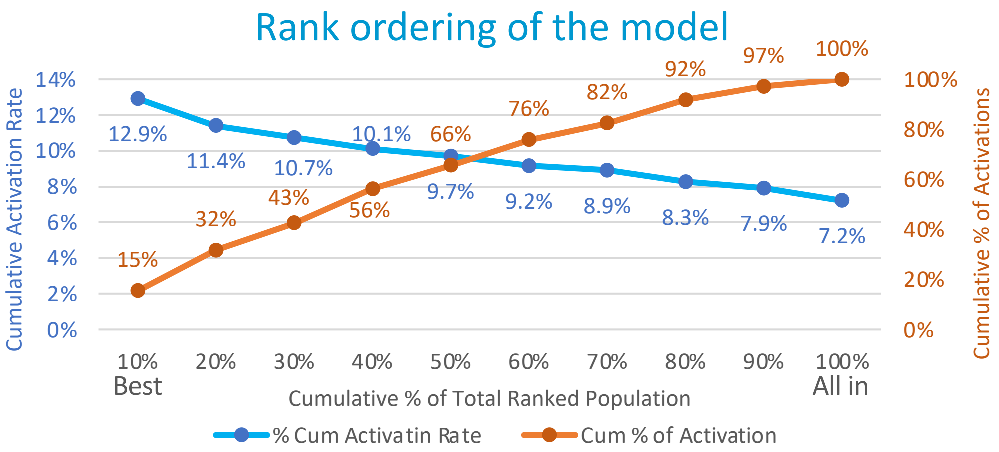
Pradeep Arkachar

February 2021

Executive Summary

Objectives

- Develop probability of activation per click model to optimize/maximize the user activation rate arising from the Google Adwords impressions
- Recommend strategy to improve marketing ROI
- Propose engineering architecture for model deployment and discuss limitations



The chart displays the ranking power of the model by showing the cumulative activation rate from the best 10% (left) to 100% of the population (right)

Recommendations

- Developed a probability of activation model that can be used to rank order customers for an efficient marketing dialer strategy shown in the chart above
- This framework can be further enhanced by combining with profitability model to go after the most profitable and converting customers
- Proposed a cloud-based engineering architecture

Insights

- 83% of the customers seek debt settlement - 61% via mobile and 21% via Desktop (higher activation)
- 98% of the customers are not prime - 50/50 split city vs non-city
- High converters in GA, PA, MD and CO; Philadelphia, Honolulu and Denver
- Cash loans have lowest conversion rate (3.8%)

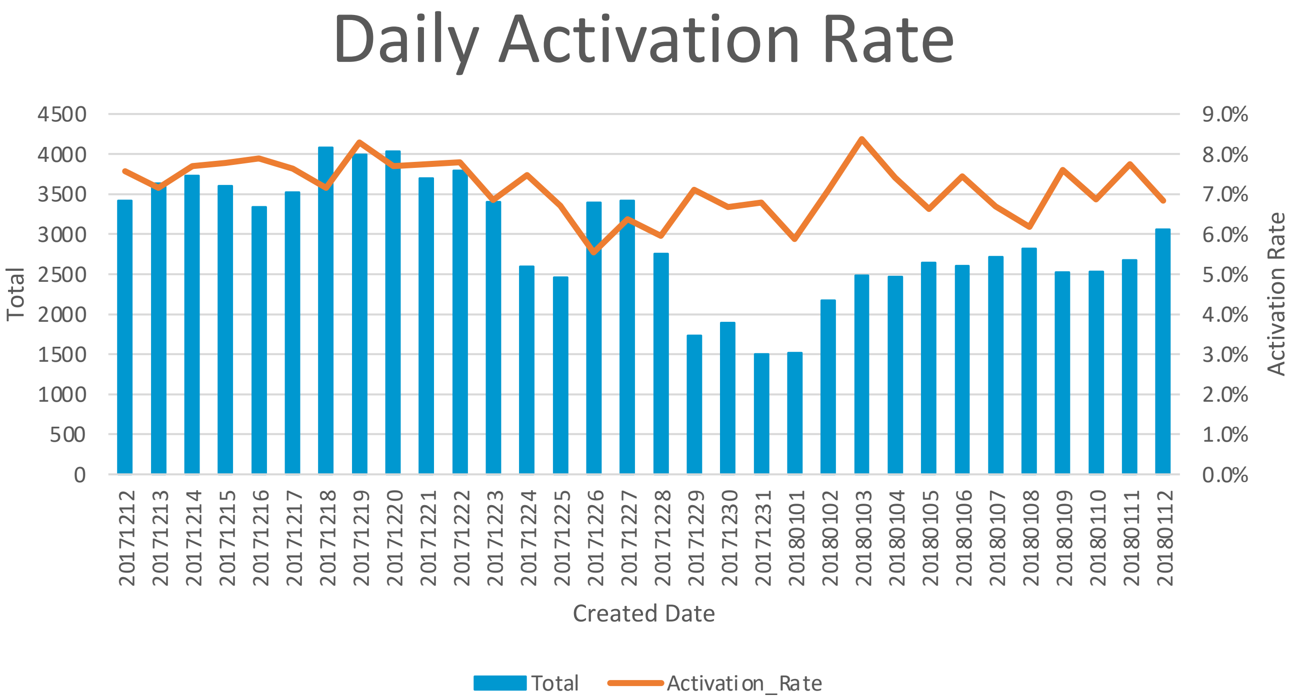
Model Development

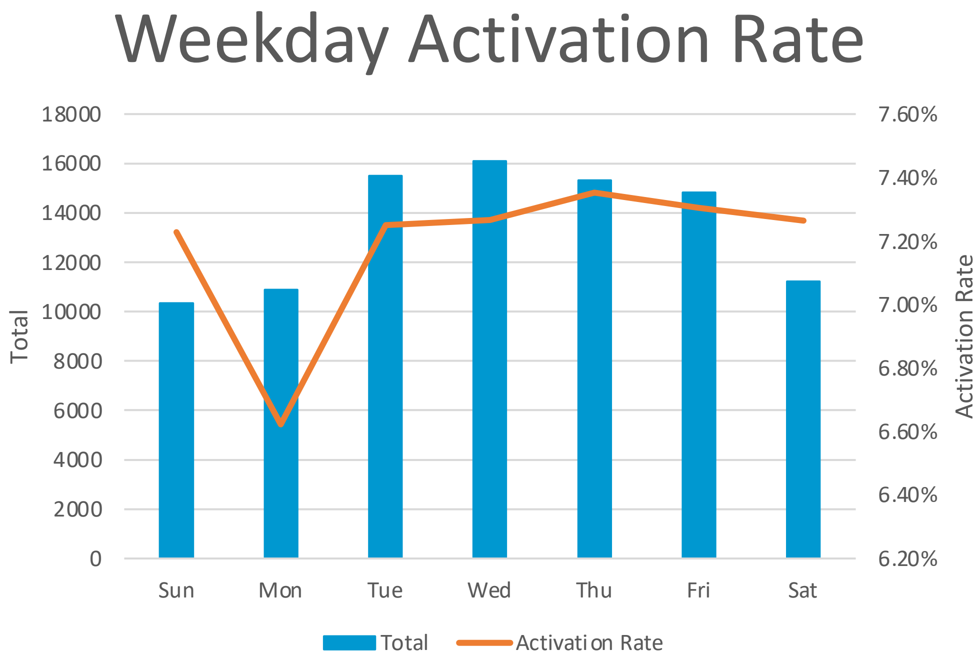
Data Summary

Total	Activated	Activation Rate
94,194	6,787	7.21%

Variable name	Total Missing	% Missing
category_debt_type	9859	10.5%
location_in_query	15	0.0%
campaign_state	6	0.0%
is_activated (target)	0	0.0%
is_hardship	0	0.0%
is_prime	0	0.0%
in_city	0	0.0%
platform	0	0.0%
created_date	0	0.0%

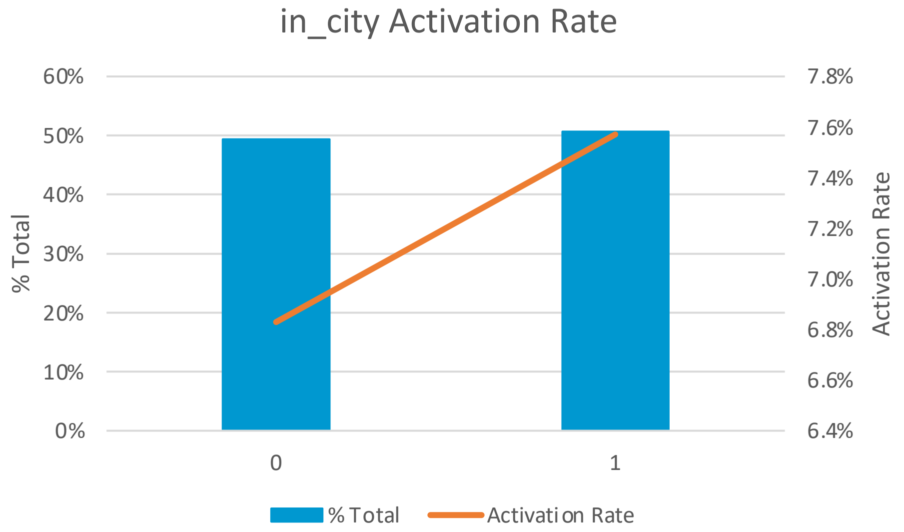
Daily and Weekly Seasonality

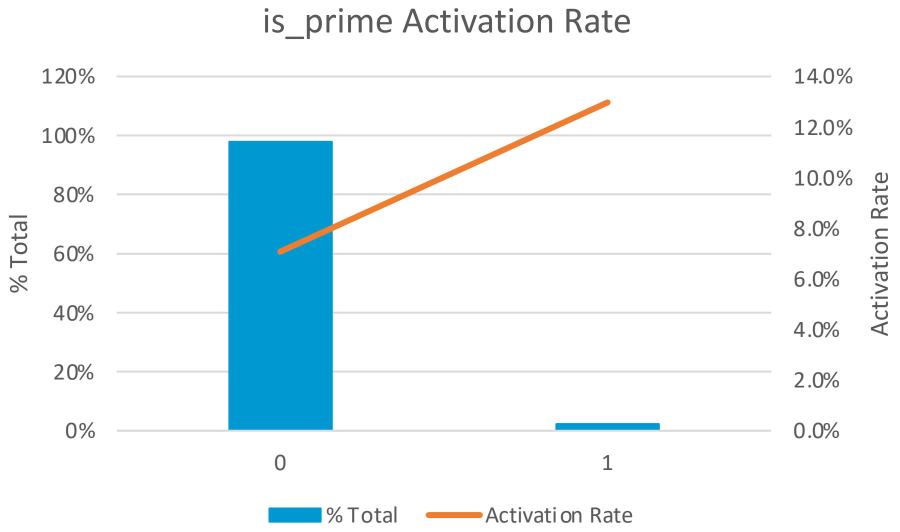


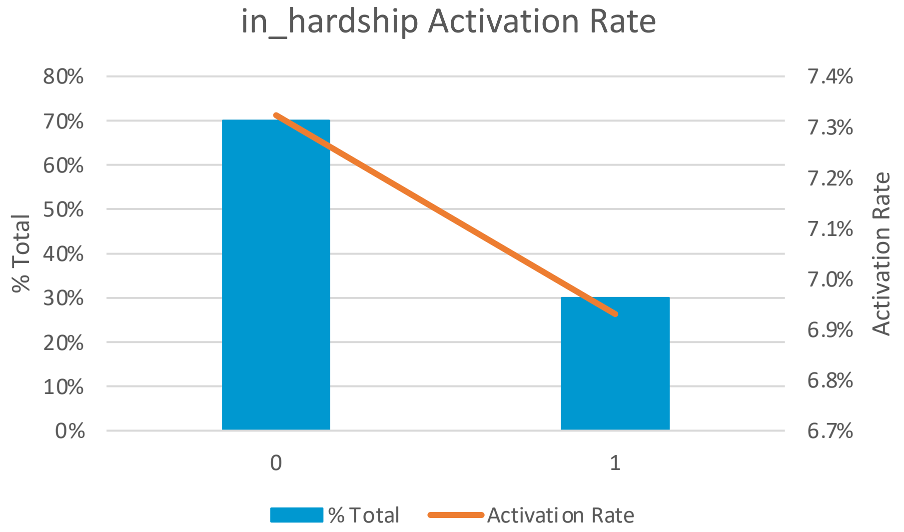


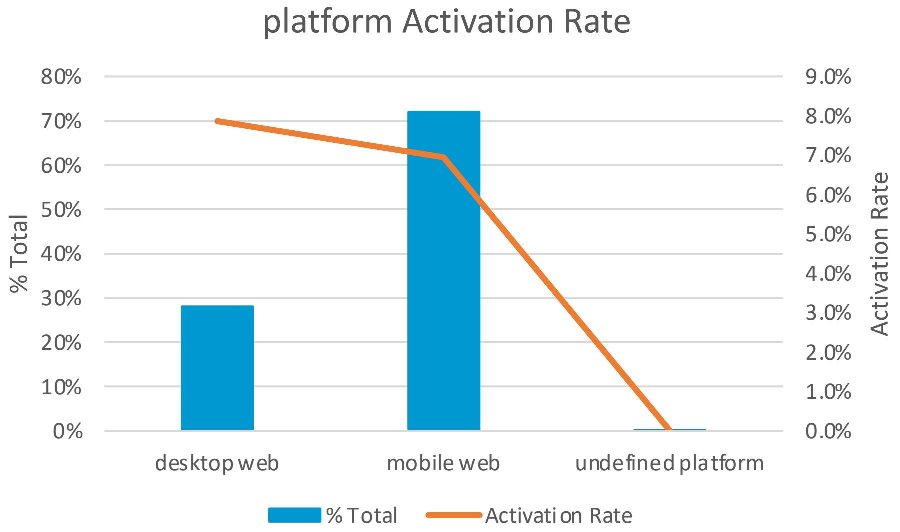
- Mondays showed lower activation rate than other weekdays
- However, given that the data only spans for a month (4 data points per weekday) and the lower activity during the last week of the year, this feature was dropped from consideration
- With more data points, assessment can be made if there is weekly and monthly seasonality

Activation Rates by features

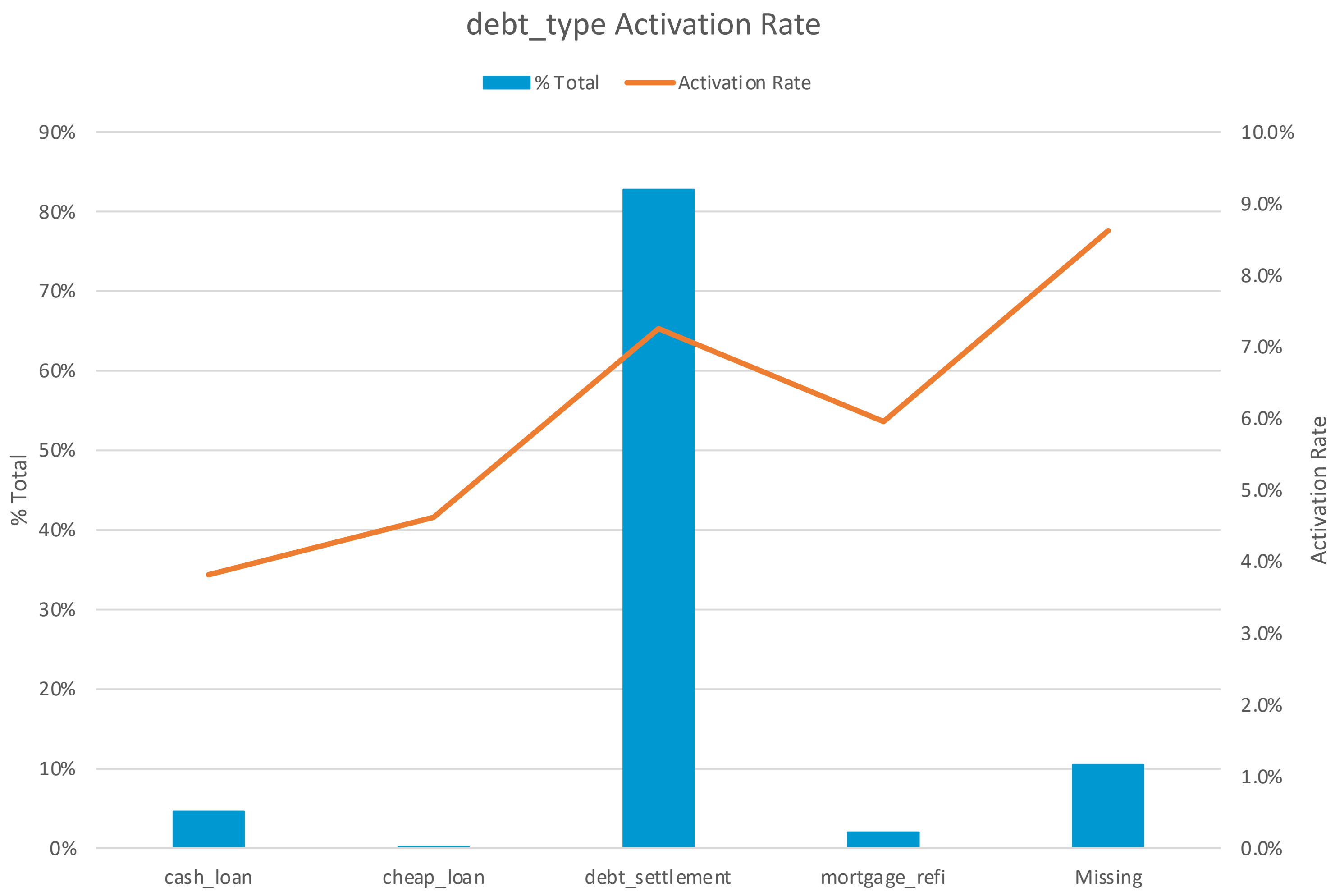




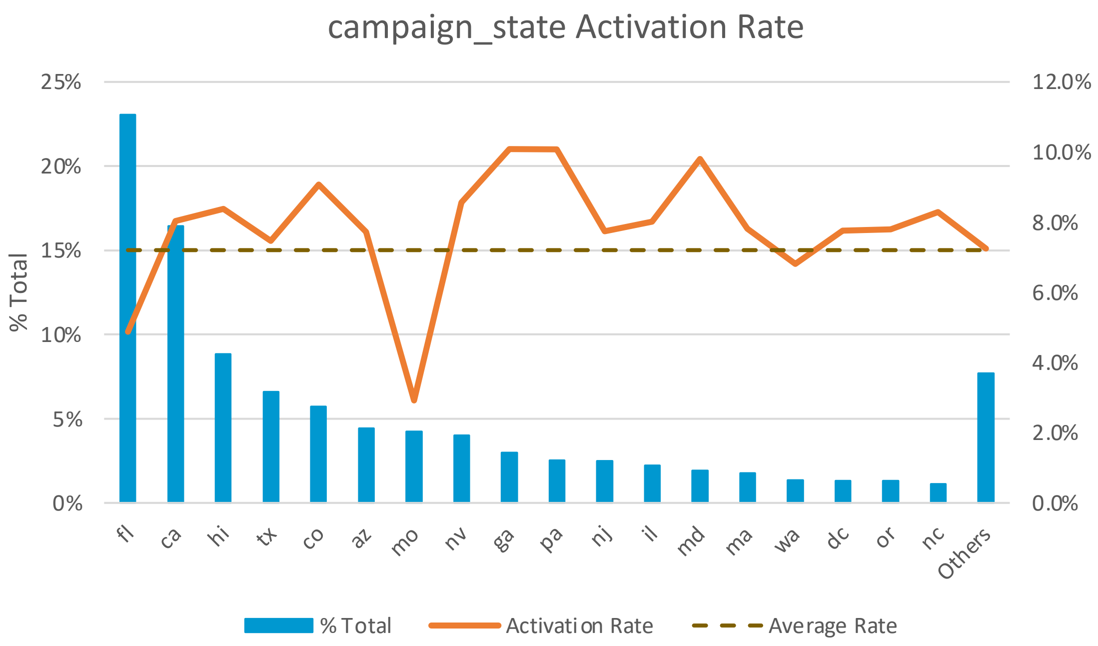


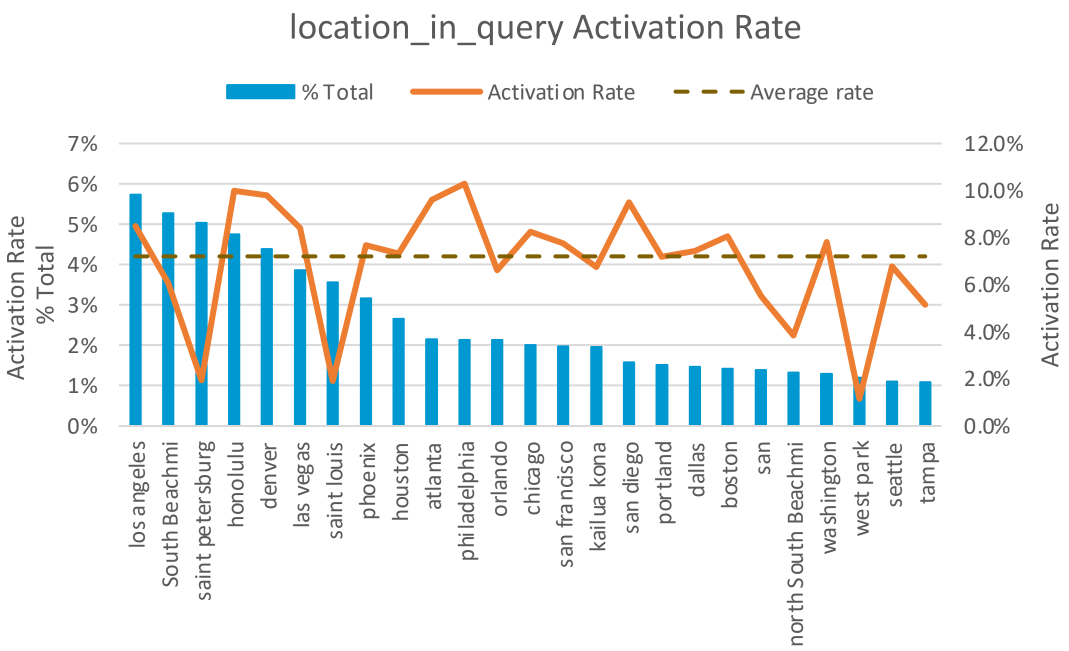


Activation Rates by Debt Type



Activation Rates by State and Location





- The states and locations with total counts less than 1000 were grouped together
- High conversion - GA, PA, MD and CO; Philadelphia, Honolulu and Denver
- Low conversion – MO; Saint Petersburg, West Park and Saint Louis

Modeling Consideration

Data

- 12/12/2017 to 01/12/2018 (~ 1 month)
- Model development data – first 3 weeks (~ 75%) train/test 80%/20%
- Out of time validation – last week (~ 25%)

Feature Engineering

- Location, State – Group categories < 25 Counts into ‘Others’
- Missing values of Debt Category was its own group
- N-1 dummy variables for each categorical variable

Algorithm

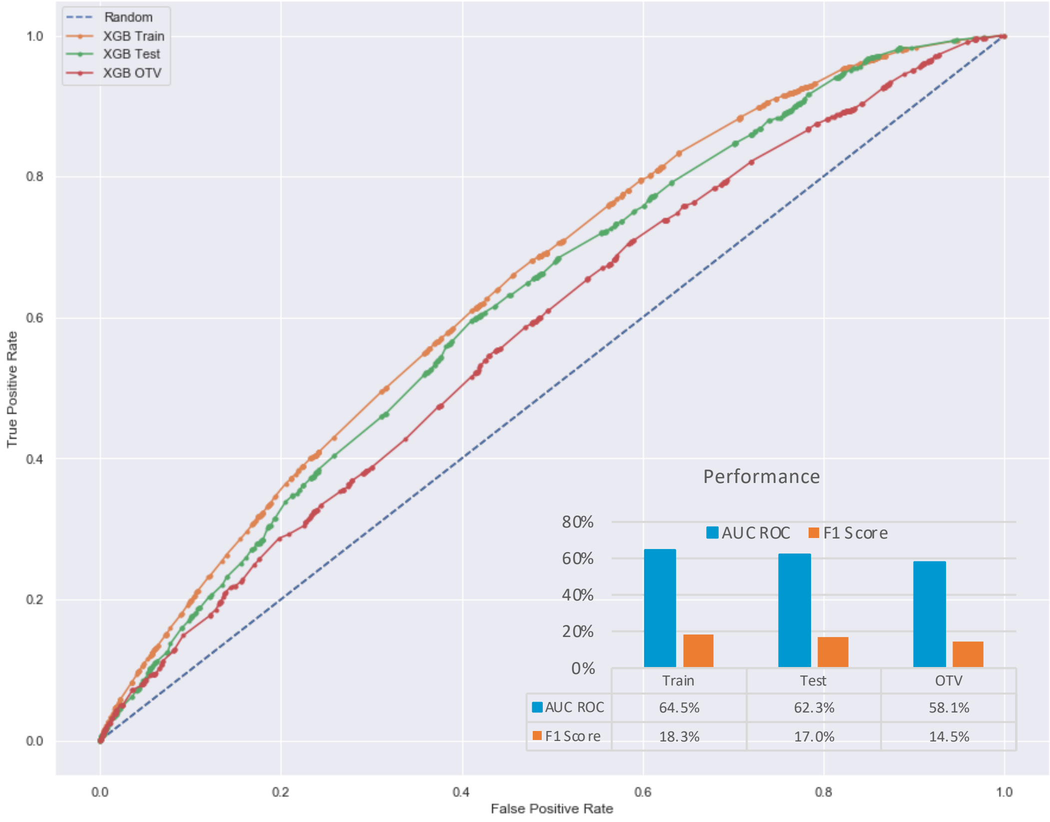
- XGBoost algorithm was chosen because of its fast development and outperformance, while offering some transparency

Validation

- K-Fold Cross validation and hyperparameter tuning was performed to avoid overfitting
- Expected VS actuals plot showed that the model fit the actual data well

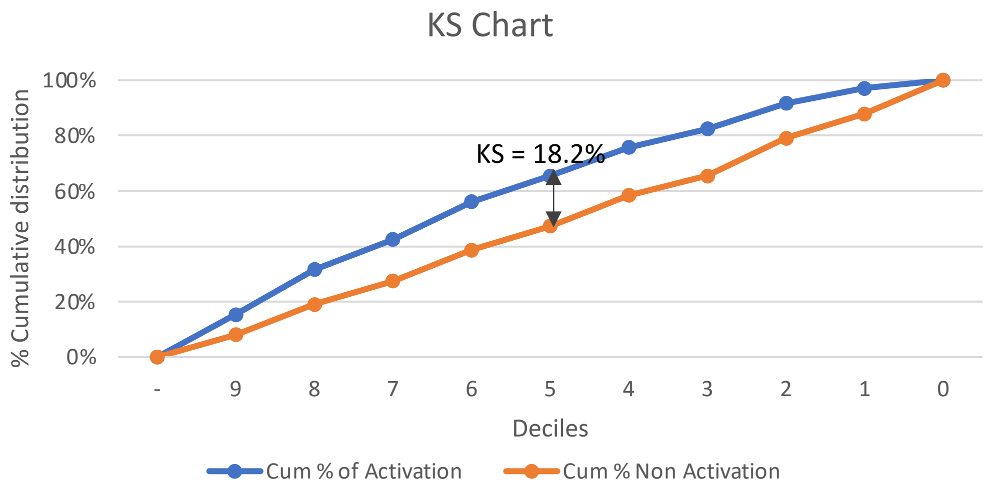
Modeling Evaluation

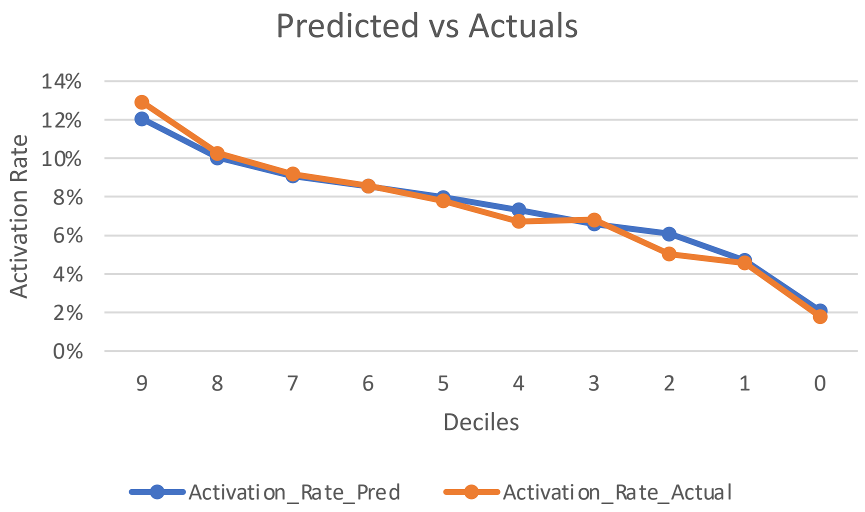
- As seen in the distribution chart below, as well as the AUC and F1 scores, the model is weak in terms of separating activations from non-activations, however, it rank orders properly as demonstrated by the rank ordering table on the next slide



Rank Ordering Table

group_10	Activation Probability Range	Total	% Cum Dist	Total Predicted Activation	Total Actual Activation	Activation_Rate_Pred	Activation_Rate_Actual	% Cum Activatin Rate	% of Activation	% of Non-activation	Cum % of Activation	Cum % Non Activation	Difference
(Best) 9	(0.105, 1]	8082	8.6%	974	1044	12.05%	12.92%	12.92%	15.4%	8.1%	15.4%	8.1%	7.3%
8	(0.0956, 0.105]	10748	20.0%	1078	1103	10.03%	10.26%	11.40%	16.3%	11.0%	31.6%	19.1%	12.5%
7	(0.0866, 0.0956]	8064	28.6%	733	740	9.08%	9.18%	10.73%	10.9%	8.4%	42.5%	27.5%	15.1%
6	(0.0826, 0.0866]	10781	40.0%	923	924	8.56%	8.57%	10.12%	13.6%	11.3%	56.2%	38.7%	17.4%
5	(0.0769, 0.0826]	8151	48.7%	649	635	7.97%	7.79%	9.70%	9.4%	8.6%	65.5%	47.3%	18.2%
4	(0.0694, 0.0769]	10383	59.7%	761	698	7.33%	6.72%	9.15%	10.3%	11.1%	75.8%	58.4%	17.4%
3	(0.0621, 0.0694]	6613	66.7%	436	450	6.60%	6.80%	8.90%	6.6%	7.1%	82.4%	65.5%	16.9%
2	(0.0523, 0.0621]	12516	80.0%	760	631	6.07%	5.04%	8.26%	9.3%	13.6%	91.7%	79.1%	12.6%
1	(0.0331, 0.0523]	8111	88.6%	381	370	4.70%	4.56%	7.90%	5.5%	8.9%	97.2%	87.9%	9.2%
(Worst) 0	(0.0, 0.0331]	10745	100.0%	224	192	2.08%	1.79%	7.21%	2.8%	12.1%	100.0%	100.0%	0.0%
Total		**94194**		**6920**	**6787**	**7.35%**	**7.21%**					*KS*	***18.2%***





Comparing Algorithms and Sampling Methods

- Given the imbalance between the minority and majority class, the following algorithms and sampling methods were considered improve performance

Algorithms

- XGBoost
- Random Forest
- LGBM
- CatBoost

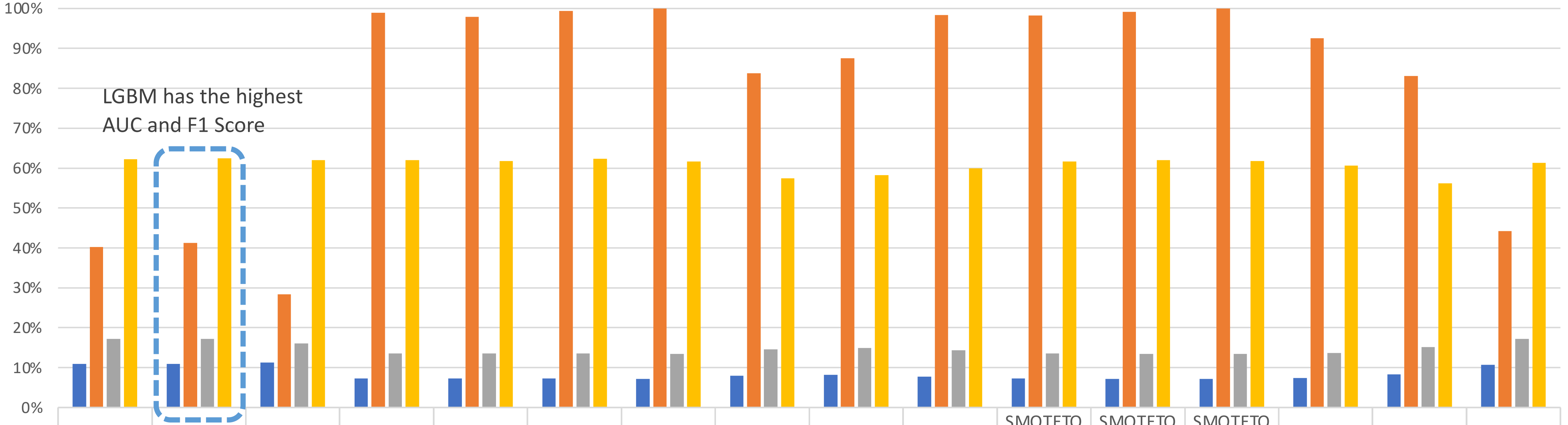
Sampling Methods

- No Sampling
- SMOTE (Oversampling Minority)
- SMOTE ENN (Hybrid)
- SMOTE Tomek (Hybrid)

- However, the performance of the algorithms were very similar in terms of AUC ROC
- The sampling methods also didn’t have any significant improvement in the performance
- The chart on the next slide shows the comparison between different algorithms
- A probability threshold of 0.09+ was set for activation

Algorithms and Sampling Methods

Comparison between Algorithms/Sampling methods



	None CatBoost	None LGBM	None XGBoost	None CatBoost Balanced	SMOTE CatBoost	SMOTE LGBM	SMOTE XGBoost	SMOTEENN CatBoost	SMOTEENN LGBM	SMOTEENN XGBoost	SMOTETOMEK CatBoost	SMOTETOMEK LGBM	SMOTETOMEK XGBoost	SMOTE RandomForest	SMOTEENN RandomForest	None RandomForest
Precision	10.9%	10.9%	11.2%	7.3%	7.3%	7.2%	7.2%	8.0%	8.1%	7.8%	7.3%	7.2%	7.2%	7.4%	8.3%	10.7%
Recall	40.2%	41.2%	28.4%	98.9%	97.9%	99.4%	100.0%	83.7%	87.6%	98.3%	98.3%	99.2%	100.0%	92.5%	83.1%	44.3%
F1-score	17.2%	17.2%	16.1%	13.5%	13.6%	13.5%	13.5%	14.5%	14.9%	14.4%	13.6%	13.5%	13.5%	13.6%	15.1%	17.2%
AUC-ROC	62.3%	62.5%	62.0%	62.0%	61.7%	62.3%	61.6%	57.4%	58.3%	59.9%	61.7%	62.0%	61.8%	60.7%	56.2%	61.3%

Precision Recall F1-score AUC-ROC

Proposed Next Steps To Improve Performance

- Explore additional data that can be obtained systematically
    - ➢Credit bureau data
    - ➢Social media data from third party data aggregators
- Assess trade off between model execution speed/ease of execution and performance
- Side by side with Call Center associates can provide additional insights into the key features they look for in a prospect; Bringing those features into modeling dataset will improve the prediction

Model Performance Tracking Post Implementation

- Model performance monitoring and tracking
  - ➢Periodic cadence (weekly, monthly)
  - ➢Check for population stability (PSI), rank ordering reversals, AUC, F Score
- Define and monitor KPIs – In this case activation rate
  - ➢A model developed with rigor will significantly improve activation rate thus increasing marketing ROI
- Evaluate additional data/features to improve model prediction on an ongoing basis

Engineering Architecture

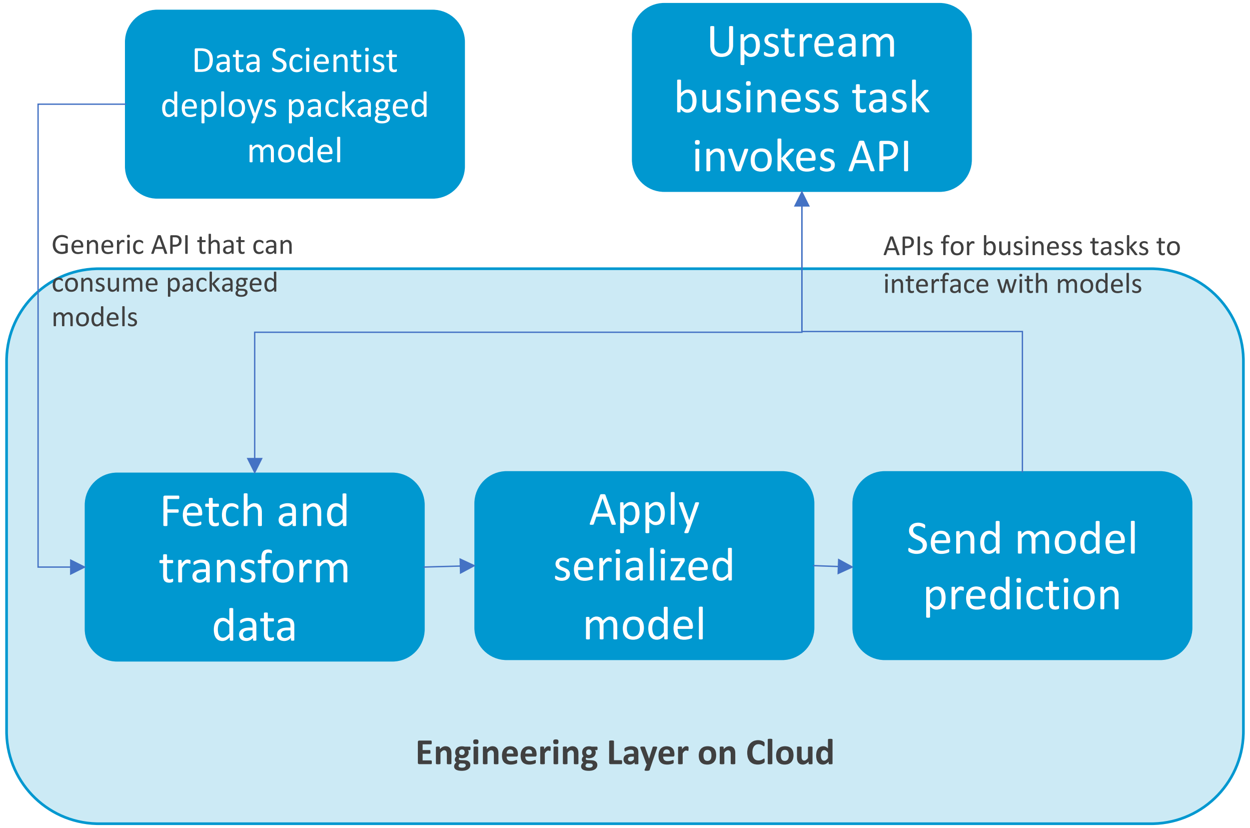
Proposed Engineering Architecture on Cloud

Workflow

1. Data scientist puts together model package – init(), pre-process(), predict()
2. Model gets deployed by generic API provided by engineering team. Many platforms such as Google, Amazon offer these services
3. Upstream process invokes Model and gets prediction

Limitations

1. In case of complex machine learning algorithm, model packaging may become hard
2. The platform may not optimally tuned for the workload and will likely be hard to tune to a specific use-case. If this is critical, then bringing some parts of the service in-house would be a part of the solution

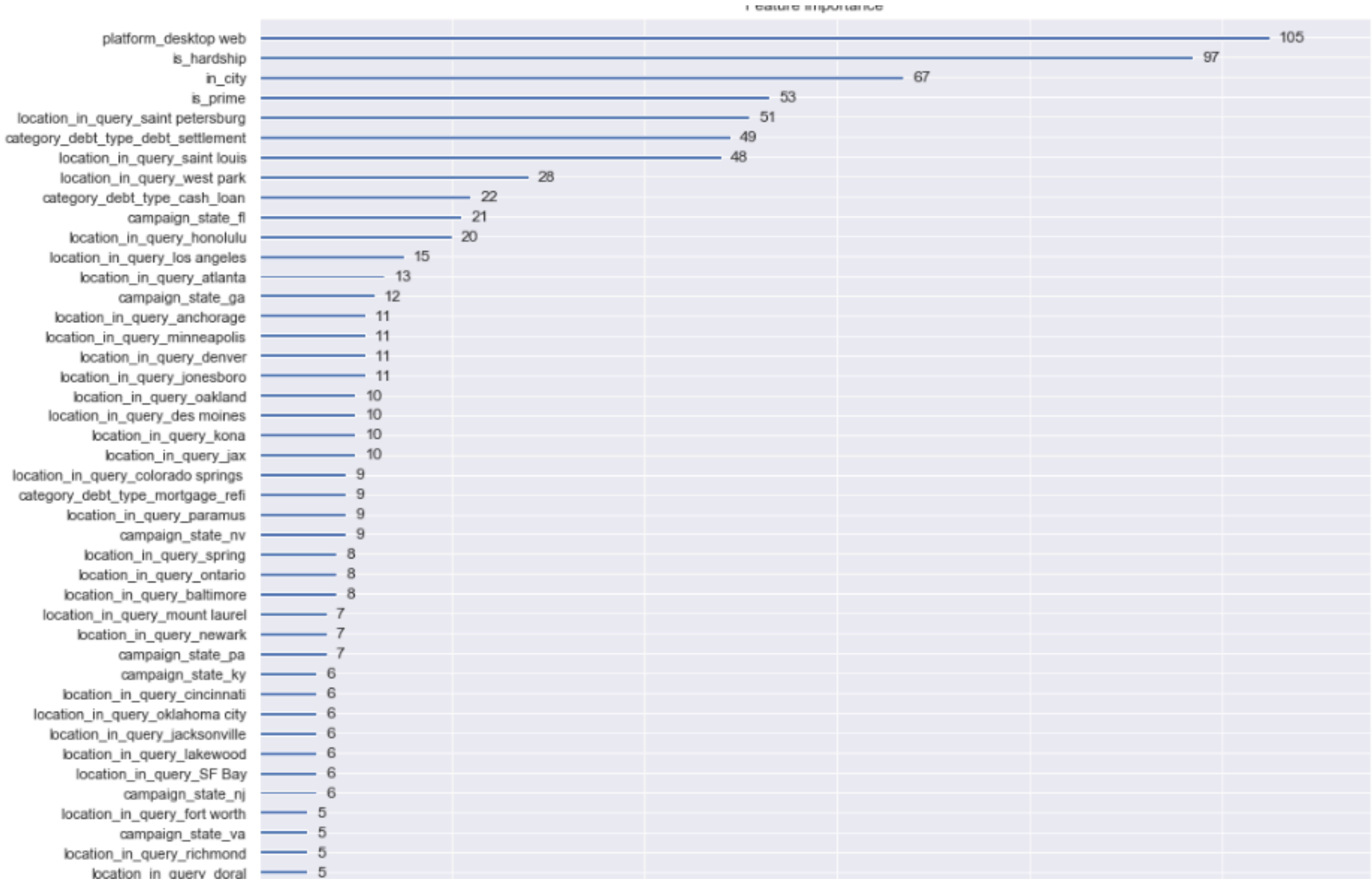


- Scalability
- Reproducibility
- Modular

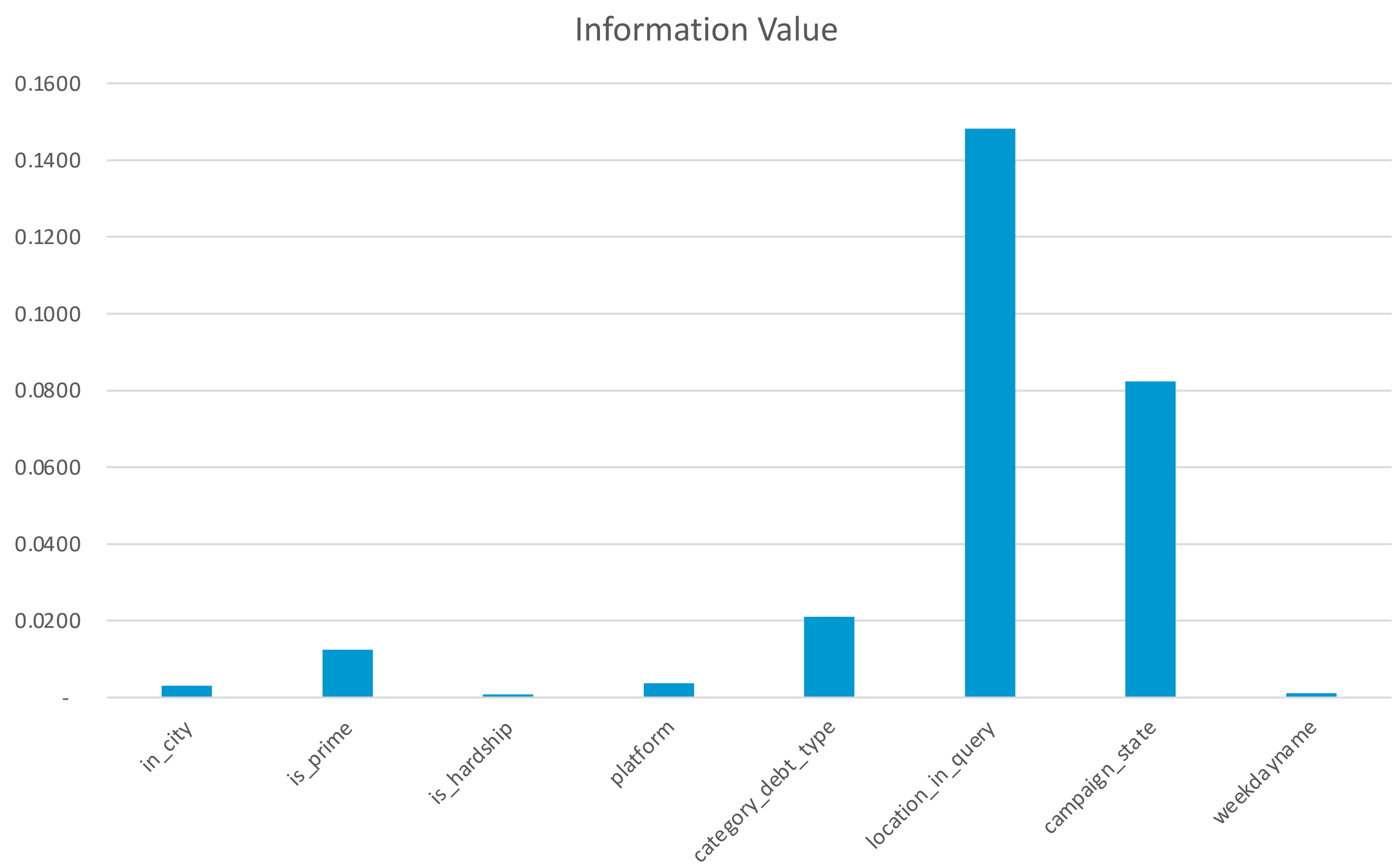
- Version control
- Automation
- Testability

Appendix

Feature Importance (Partial list)



Information Values



- State and Location have IV indicating that targeting certain geographies can lead to high conversion rate