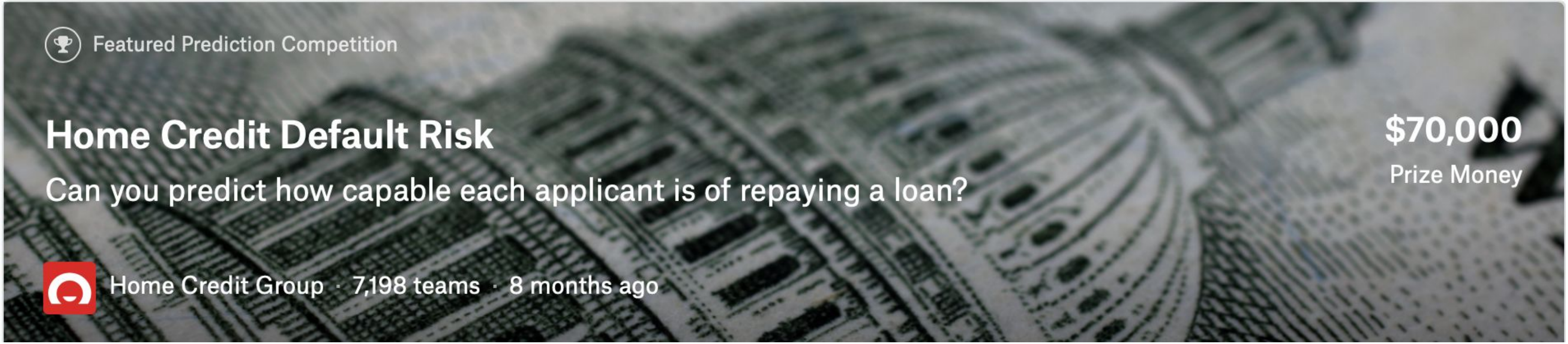
Home Credit Default Risk

Natus Vincere

Sumer Singh
Hemanth Dandu
Sushanth Kathirvelu

The Challenge

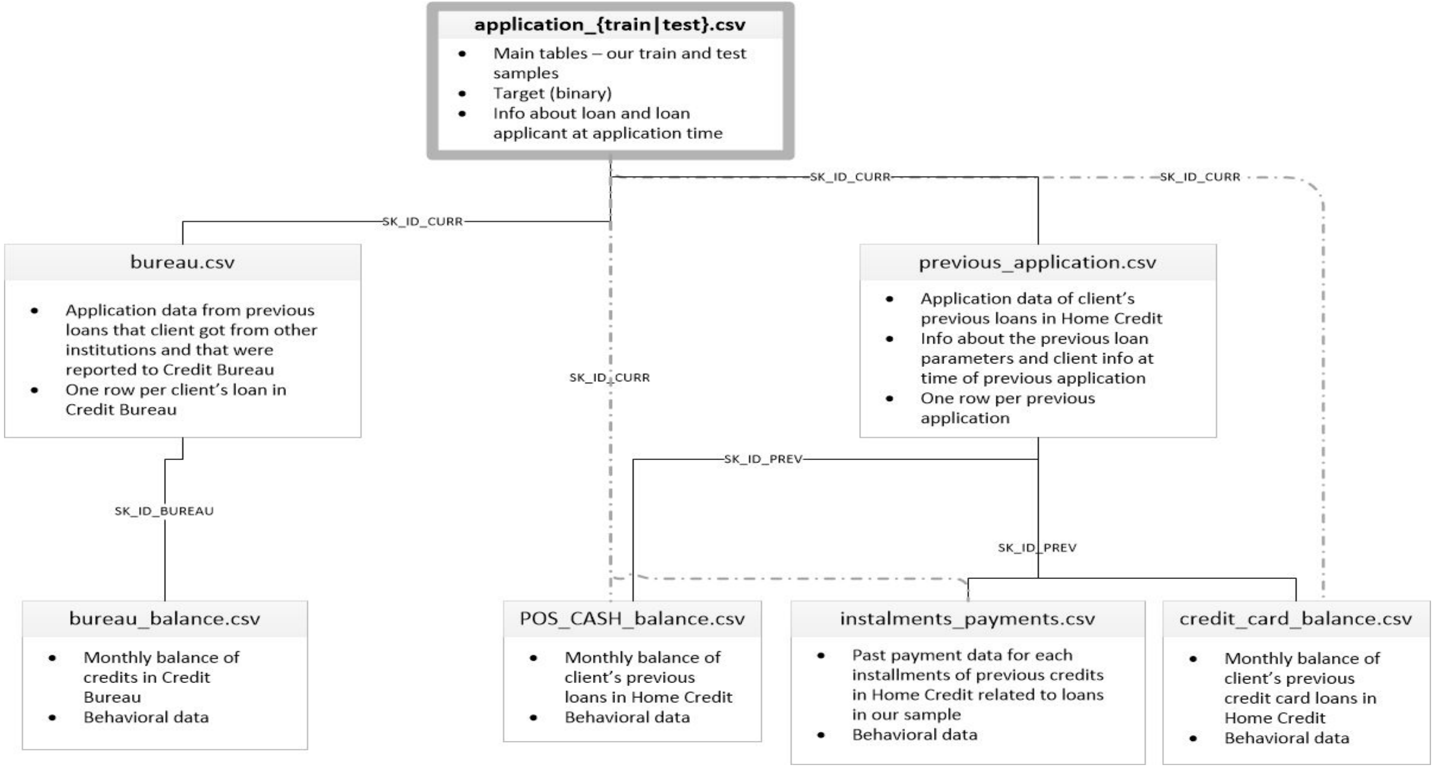
- Predict if an applicant is capable of repaying a loan
- Evaluation Criteria: Area under the ROC curve between the predicted probability and the observed target
- https://www.kaggle.com/c/home-credit-default-risk



The Data

- Application_train data (307,511 * 122)
- Application_test data (48,744 * 121)
- Bureau data (1,716,428 * 17)
    - Bureau_balance data (27,299,925 * 3)
- Previous_application data (1,670,214 * 37)
    - Installments_payments data (13,605,401 * 8)
    - Credit_card_balance data (3,840,312 * 23)
    - POS_CASH_balance data (10,001,358 * 8)

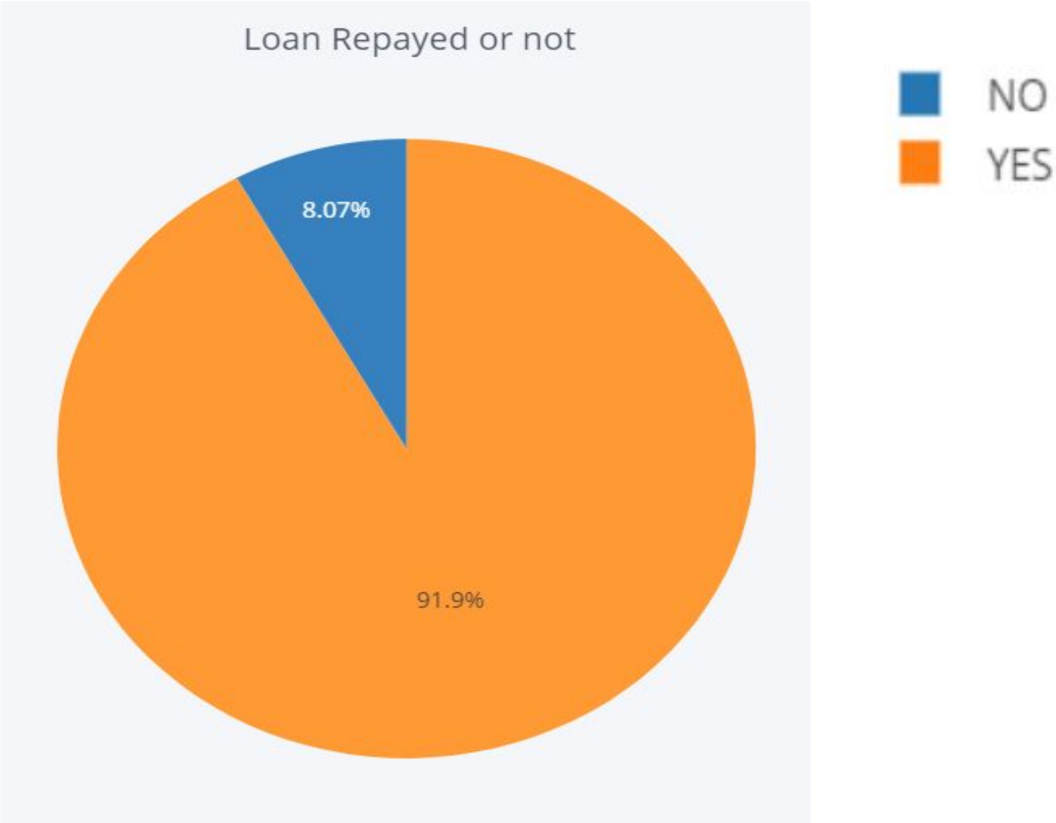
Data Model

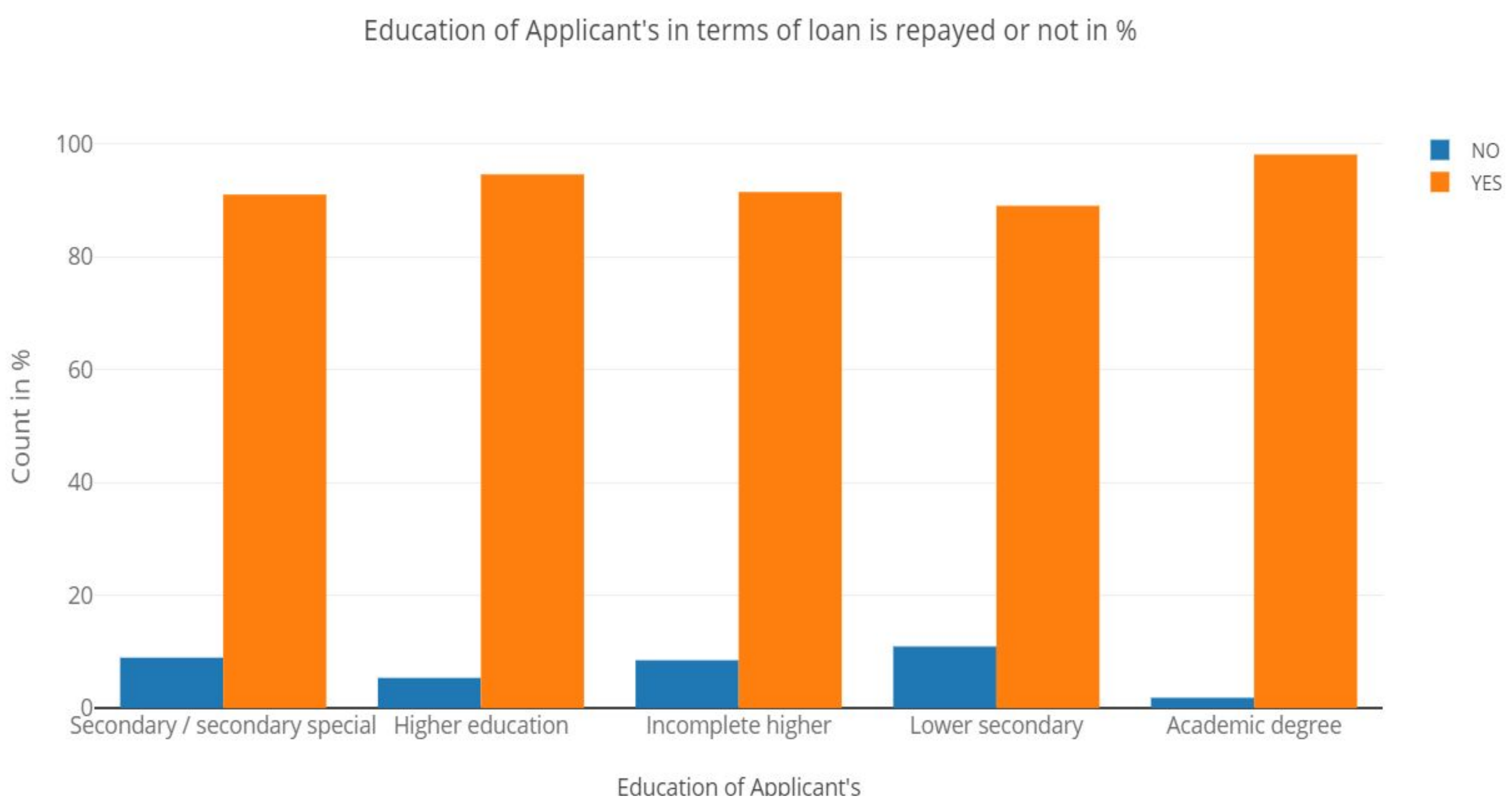


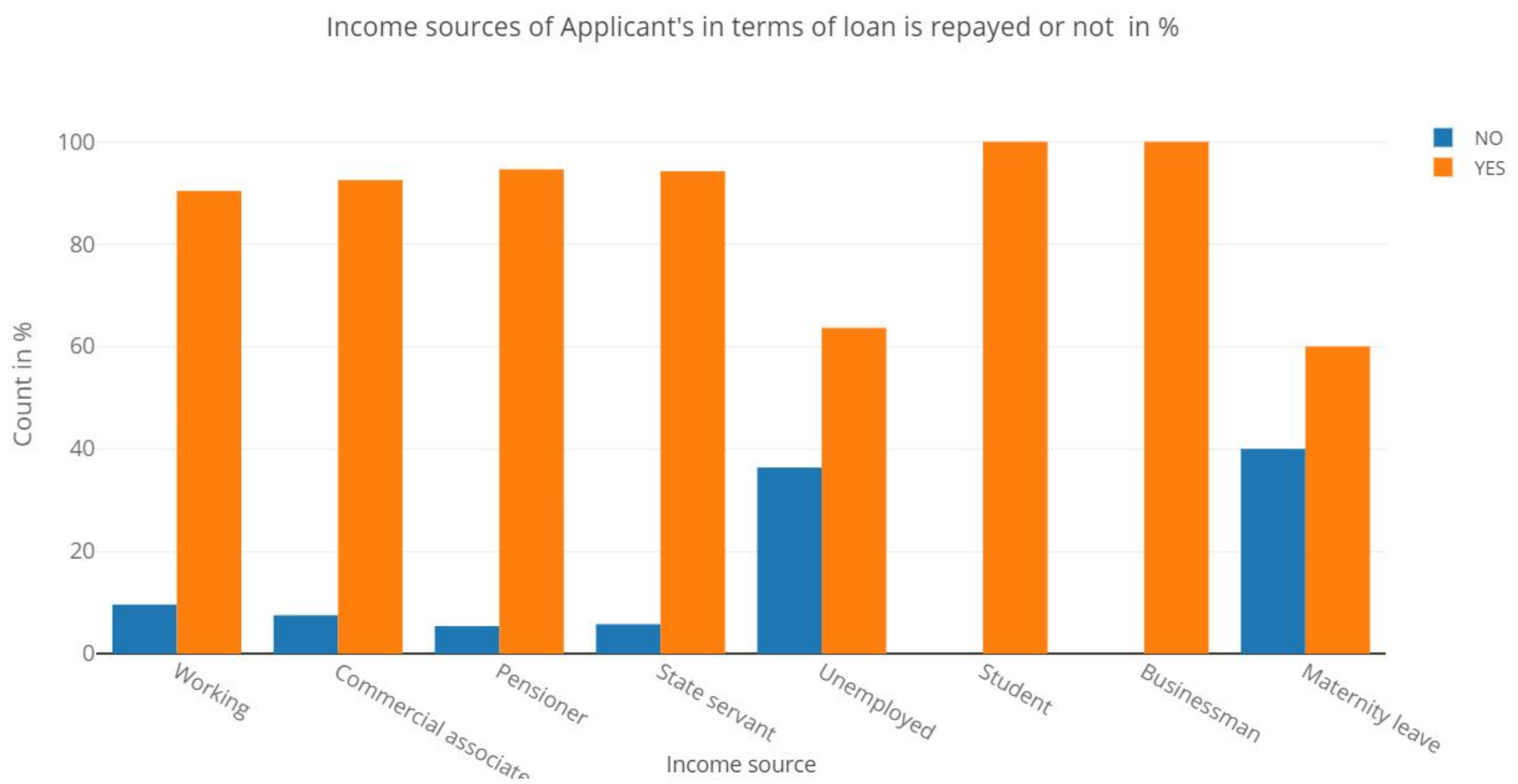
application_{train|test}.csv
Main tables – our train and test samples
Target (binary)
Info about loan and loan applicant at application time
SK_ID_CURR
SK_ID_CURR
SK_ID_CURR
SK_ID_CURR
bureau.csv
Application data from previous loans that client got from other institutions and that were reported to Credit Bureau
One row per client's loan in Credit Bureau
previous_application.csv
Application data of client's previous loans in Home Credit
Info about the previous loan parameters and client info at time of previous application
One row per previous application
SK_ID_PREV
SK_ID_BUREAU
SK_ID_PREV
bureau_balance.csv
Monthly balance of credits in Credit Bureau
Behavioral data
POS_CASH_balance.csv
Monthly balance of client's previous loans in Home Credit
Behavioral data
instalments_payments.csv
Past payment data for each installments of previous credits in Home Credit related to loans in our sample
Behavioral data
credit_card_balance.csv
Monthly balance of client's previous credit card loans in Home Credit
Behavioral data

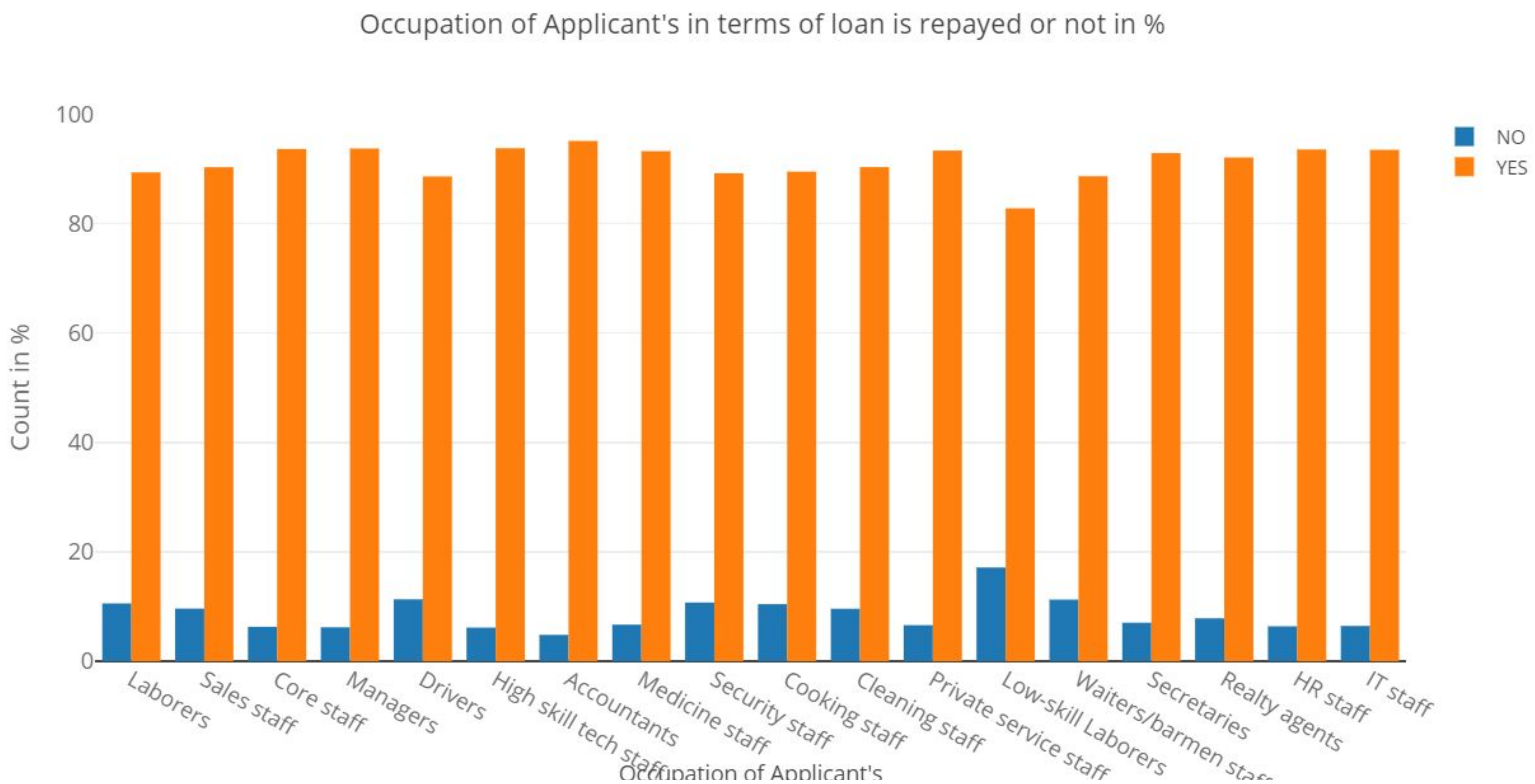
EDA

- Target Attribute Distribution
- Number of Missing Values in each column for all the data tables and its Percentage.
- Single Attribute distributions
- Distribution of Features Vs the Target

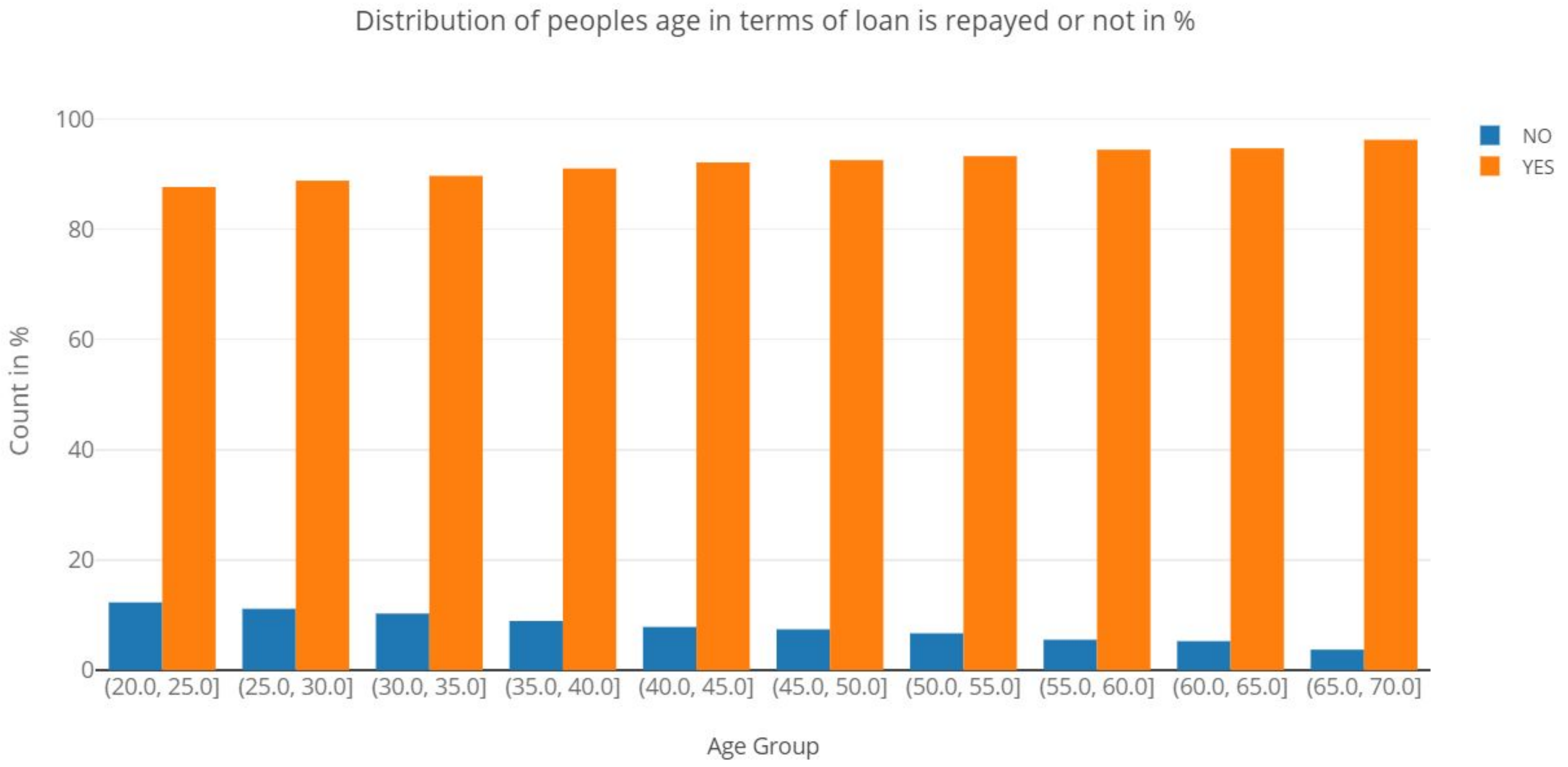

Loan Repayed or not
8.07%
91.9%
NO
YES


Education of Applicant's in terms of loan is repayed or not in %
NO
YES
Count in %
0
20
40
60
80
100
Secondary / secondary special
Higher education
Incomplete higher
Lower secondary
Academic degree
Education of Applicant's

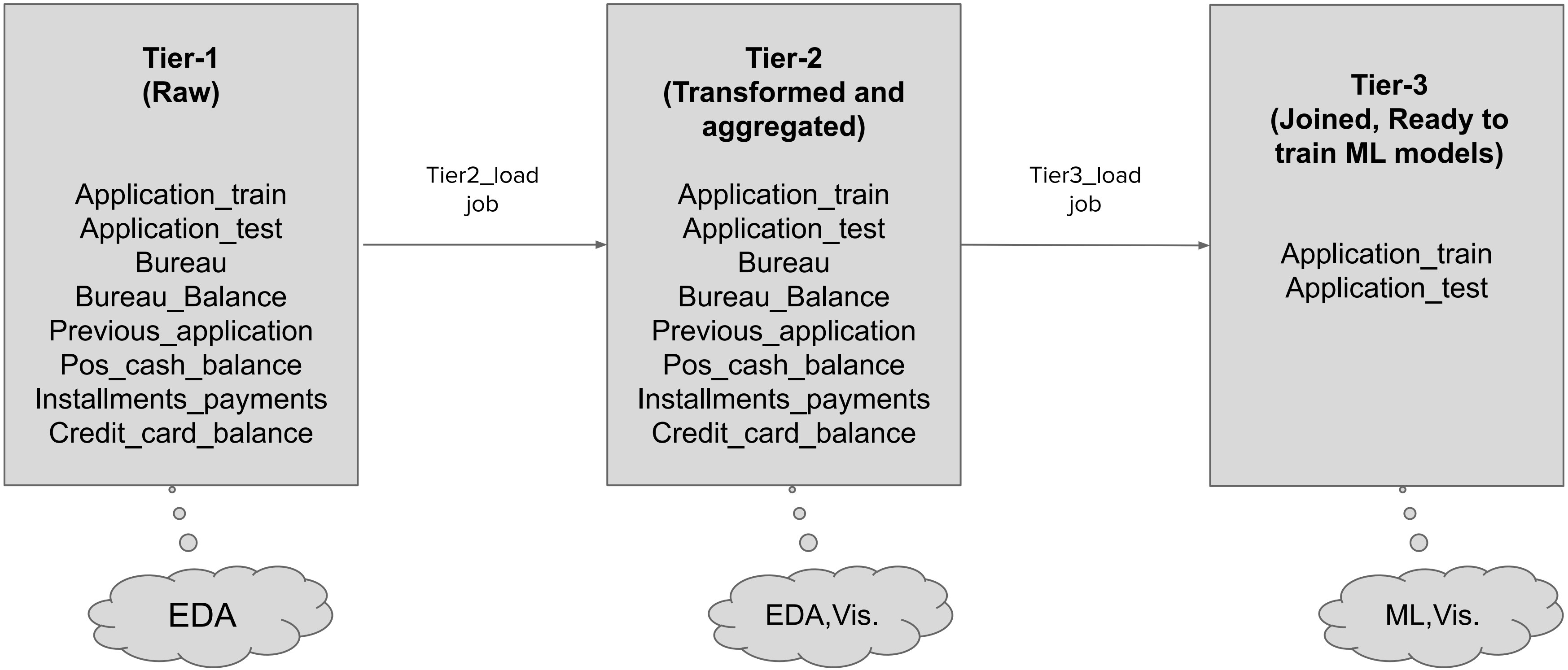

Income sources of Applicant's in terms of loan is repayed or not in %
NO
YES
Count in %
0
20
40
60
80
100
Working
Commercial associate
Pensioner
State servant
Unemployed
Student
Businessman
Maternity leave
Income source



Occupation of Applicant's in terms of loan is repayed or not in %
NO
YES
Count in %
100
80
60
40
20
0
Laborers
Sales staff
Core staff
Managers
Drivers
High skill tech staff
Accountants
Medicine staff
Security staff
Cooking staff
Cleaning staff
Private service staff
Low-skill Laborers
Waiters/barmen staff
Secretaries
Realty agents
HR staff
IT staff
Occupation of Applicant's


Distribution of peoples age in terms of loan is repayed or not in %
100
80
60
40
20
0
Count in %
NO
YES
(20.0, 25.0]
(25.0, 30.0]
(30.0, 35.0]
(35.0, 40.0]
(40.0, 45.0]
(45.0, 50.0]
(50.0, 55.0]
(55.0, 60.0]
(60.0, 65.0]
(65.0, 70.0]
Age Group

Data Load - Process Flow



ML Approaches

- Feature Engineering:
  - Manual
  - Automated

- Training models:
  - RandomForest
  - Gradient Boosting Machines(GBM)

Manual Feature Engineering

- Convert all categorical features to dummies
- Take min,max,mean and var for all numerical features that need aggregation
- Drop any constant columns
- Add domain specific features based on literature, domain expertise etc..

Automated Feature Engineering

Featuretools is a framework to perform automated feature engineering

- It works best with relational datasets.
- Automatically aggregates and transforms features to create new features.

```
In [3]: customers_df = data["customers"]

In [4]: customers_df
Out[4]:
   customer_id zip_code           join_date date_of_birth
0            1    60091 2011-04-17 10:48:33    1994-07-18
1            2    13244 2012-04-15 23:31:04    1986-08-18
2            3    13244 2011-08-13 15:42:34    2003-11-21
3            4    60091 2011-04-08 20:08:14    2006-08-15
4            5    60091 2010-07-17 05:27:50    1984-07-28

In [5]: sessions_df = data["sessions"]

In [6]: sessions_df.sample(5)
Out[6]:
    session_id  customer_id   device       session_start
13          14            1   tablet 2014-01-01 03:28:00
6            7            3   tablet 2014-01-01 01:39:40
1            2            5   mobile 2014-01-01 00:17:20
28          29            1   mobile 2014-01-01 07:10:05
24          25            3  desktop 2014-01-01 05:59:40

In [7]: transactions_df = data["transactions"]

In [8]: transactions_df.sample(5)
Out[8]:
     transaction_id  session_id    transaction_time product_id  amount
74              232           5 2014-01-01 01:20:10          1  139.20
231              27          17 2014-01-01 04:10:15          2   90.79
434              36          31 2014-01-01 07:50:10          3   62.35
420              56          30 2014-01-01 07:35:00          3   72.70
54              444           4 2014-01-01 00:58:30          4   43.59
```

Feature Tools

```
In [12]: feature_matrix_customers
Out[12]:
```

	zip_code	COUNT(sessions)	NUM_UNIQUE(sessions.device)	MODE(sessions.device)	SUM(transactions.amount)	STD(transactions.amount)	MAX(transactions.amount)	SKEW(transact
customer_id								
1	60091	8	3	mobile	9025.62	40.442059	139.43	
2	13244	7	3	desktop	7200.28	37.705178	146.81	
3	13244	6	3	desktop	6236.62	43.683296	149.15	
4	60091	8	3	mobile	8727.68	45.068765	149.95	
5	60091	6	3	mobile	6349.66	44.095630	149.02	

Feature Tools

customer_id	MEAN(sessions.SUM(transactions.amount))
1	1128.202500
2	1028.611429
3	1039.436667
4	1090.960000
5	1058.276667

For each customer this feature

1. calculates the sum of all transaction amounts per session to get total amount per session,
2. then applies the mean to the total amounts across multiple sessions to identify the *average amount spent per session*

customer_id	MODE(sessions.HOUR(session_start))
1	6
2	3
3	5
4	1
5	0

For each customer this feature calculates

1. The hour of the day each of his or her sessions started, then
2. uses the statistical function mode to identify the most common hour he or she started a session

Primitives

Aggregations :

Default: [“sum”, “std”, “max”, “skew”, “min”, “mean”, “count”, “percent_true”, “n_unique”, “mode”

Transformations :

Default: [“day”, “year”, “month”, “weekday”, “haversine”, “num_words”, “num_characters”]

- It is also possible to create your own custom primitives.

Features Created

We created two sets of features.

1. First set created using default primitives.
2. Second set created using a subset of primitives.
    a. Subset primitives : ['sum', 'count', 'min', 'max', 'mean', 'mode']

Features Selection

Number of features:

Manual	Default Primitives	Subset Primitives
727	1984	1171

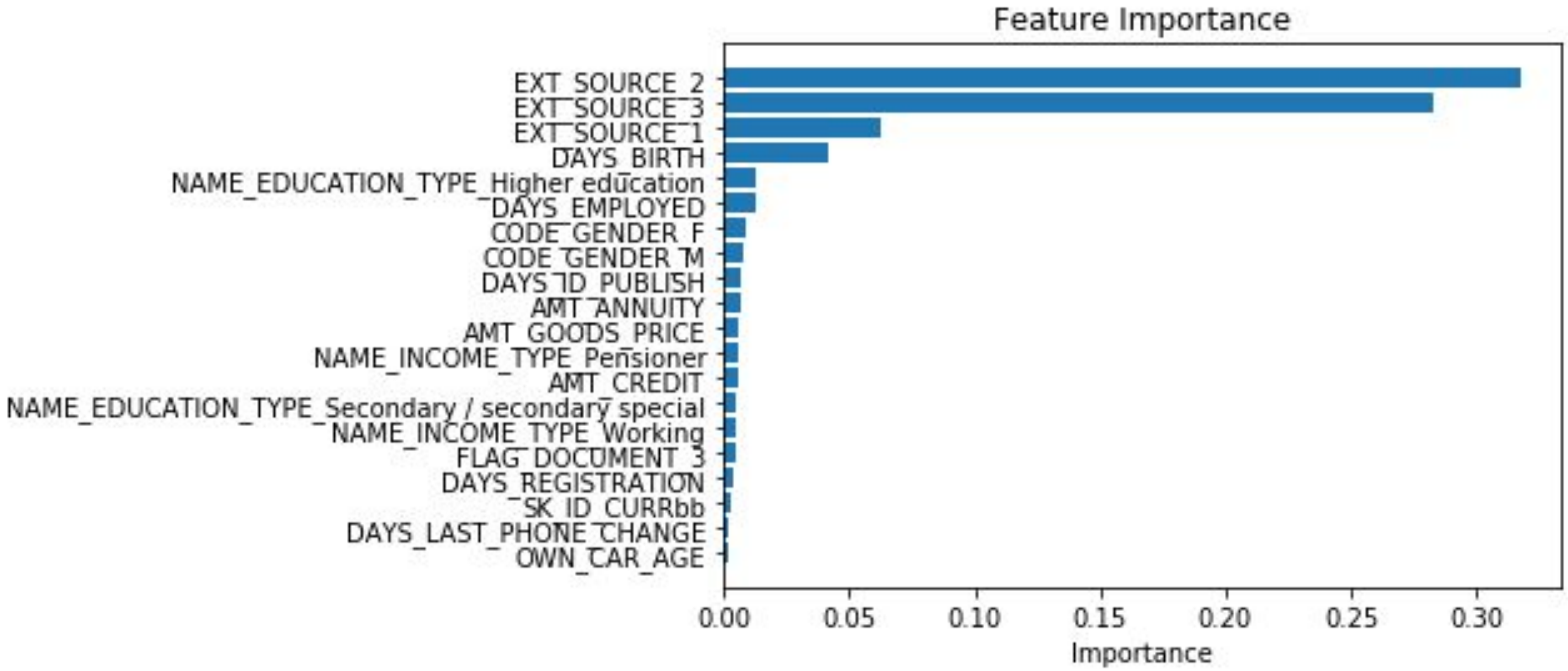
Features Selection

- Use random forest and gradient boosting to get feature importance.
- Features below a threshold are removed. Default threshold is 0.

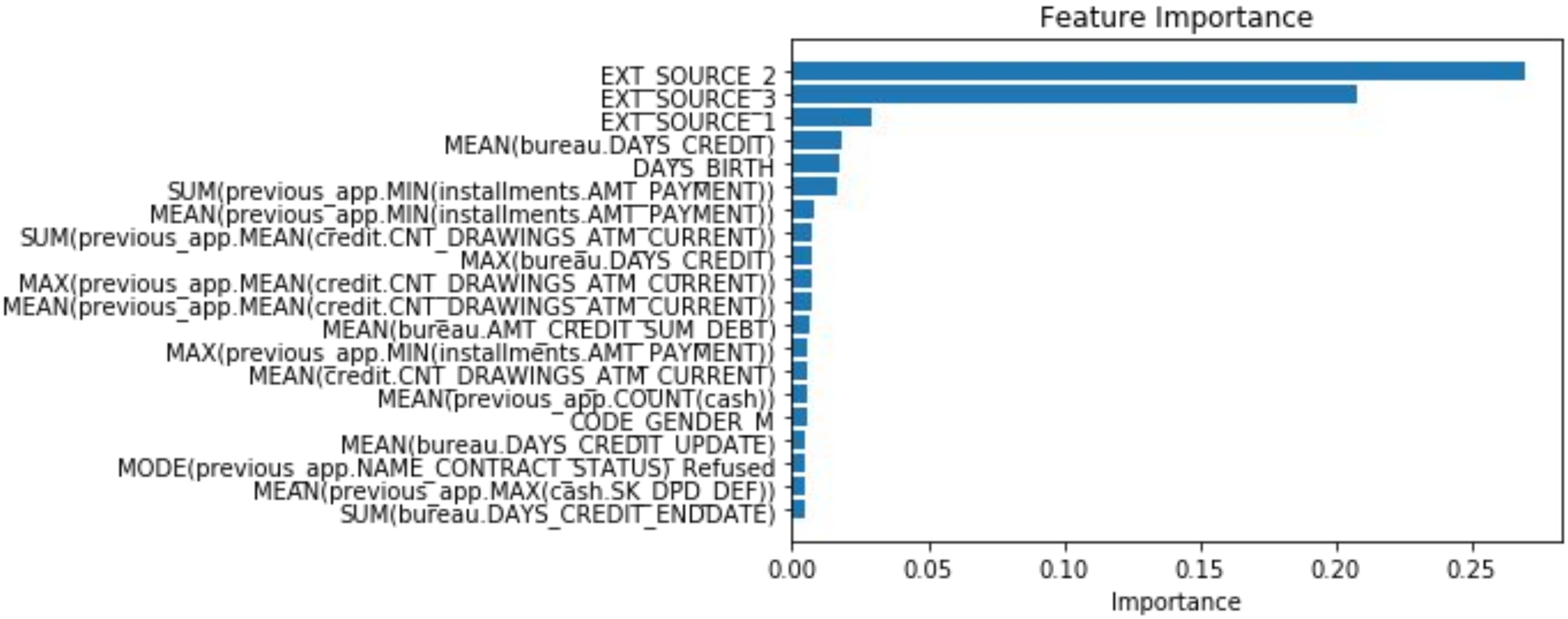
Number of features

	Manual	Default Primitives	Subset Primitives
Before	727	1984	1171
After	513	1218	879

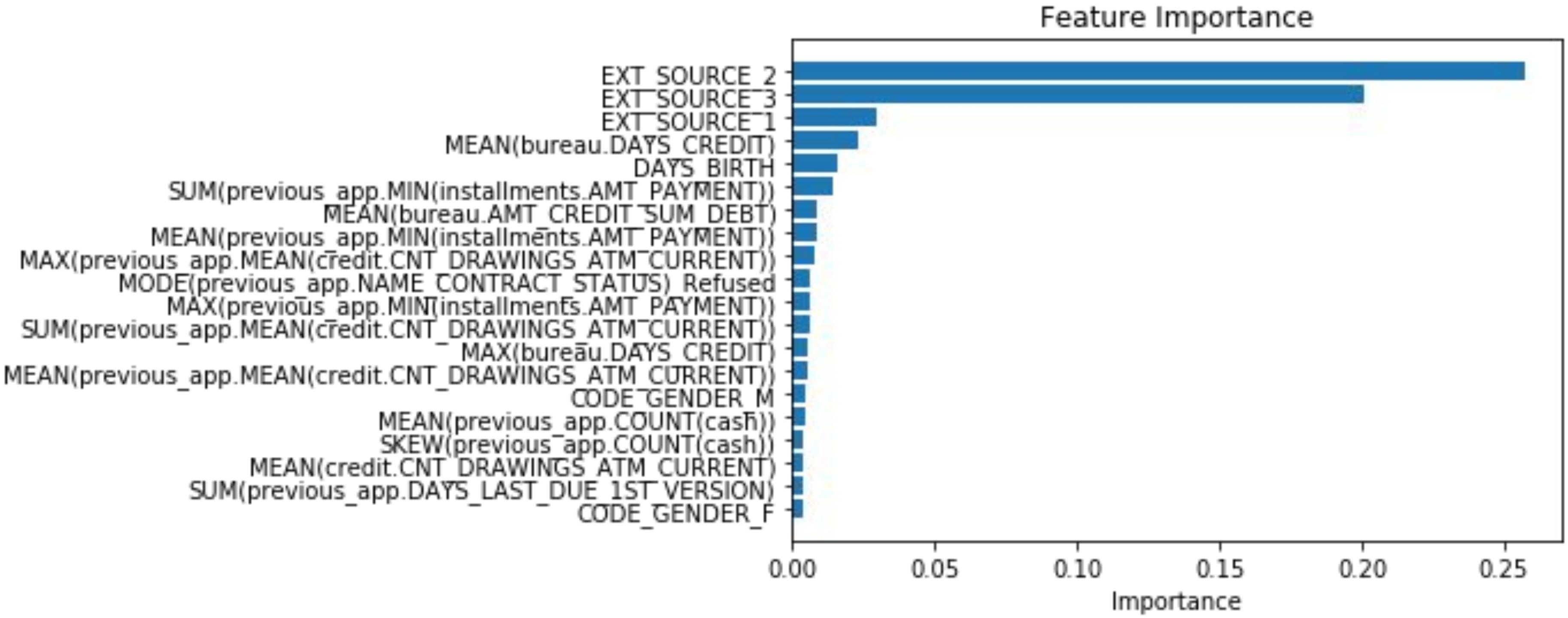
Best Features - Manual



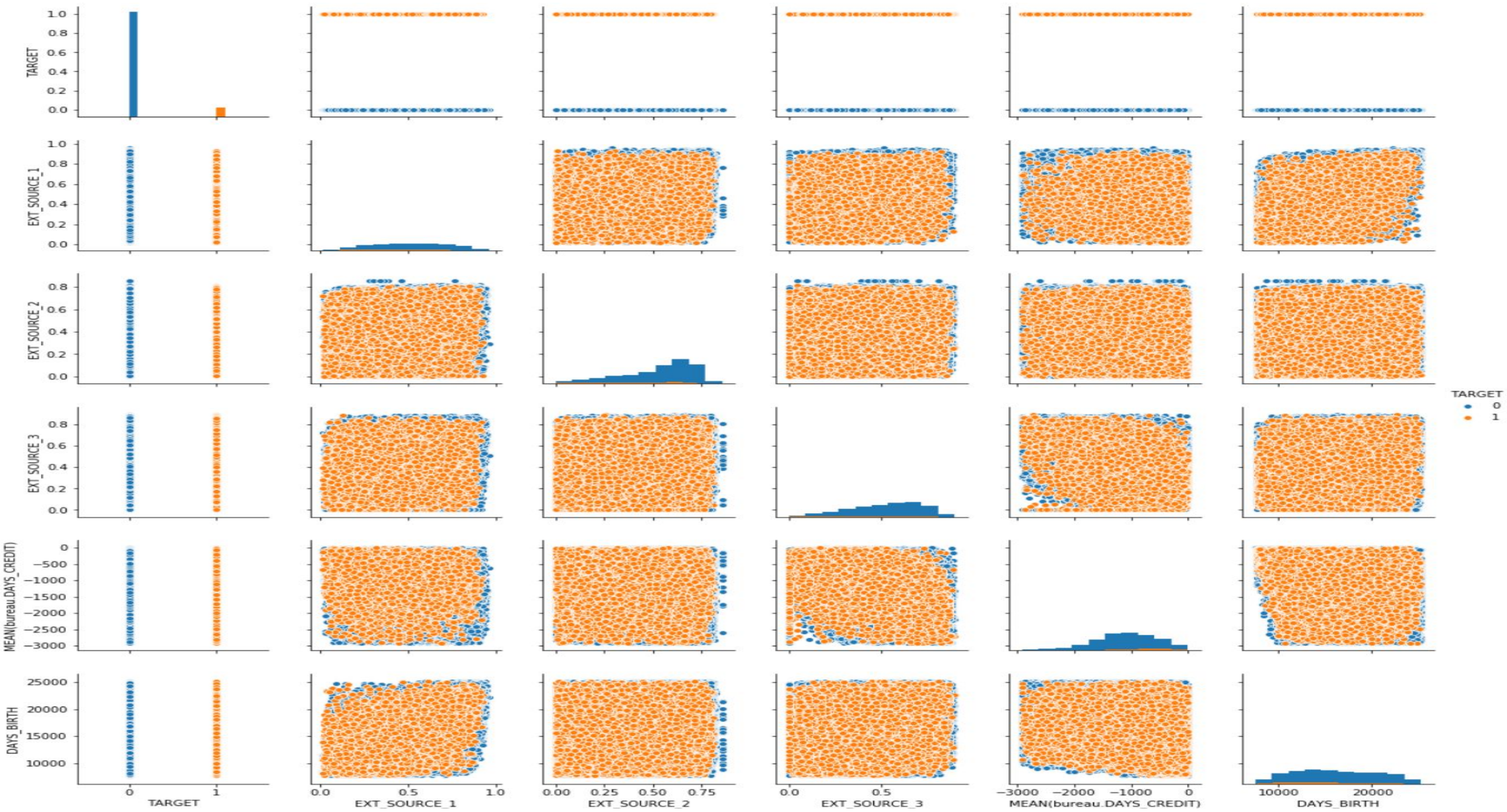
Best Features - Automatic (subset)



Best Features - Automatic (all)



Visualizing Best Features



Model

- LightGBM with Bayesian Optimized Parameters. @https://www.kaggle.com/sz8416/simple-bayesian-optimization-for-lightgbm

- Random Forest Default Parameters

```
# LightGBM parameters found by Bayesian optimization
clf = LGBMClassifier(
    nthread=4,
    n_estimators=10000,
    learning_rate=0.02,
    num_leaves=34,
    colsample_bytree=0.9497036,
    subsample=0.8715623,
    max_depth=8,
    reg_alpha=0.041545473,
    reg_lambda=0.0735294,
    min_split_gain=0.0222415,
    min_child_weight=39.3259775,
    silent=-1,
    verbose=-1, )
```

Results

Submission and Description	Private Score	Public Score
p4sub_lgbm.csv a day ago by Hemanth add submission details	0.74545	0.74480

Submission and Description	Private Score	Public Score
submission2.csv 2 days ago by sumer add submission details	0.77362	0.76951
submission.csv 3 days ago by sumer add submission details	0.77048	0.76640
subb.csv 4 days ago by sumer add submission details	0.77182	0.76962

Results

Method and features	Score
LightGBM Manual Features	0.74545
LightGBM Full automated features	0.77048
Random Forest Subset automated features	0.77182
LightGBM Subset automated features	0.77362

Kaggle best score - 0.80570

Pending Work

- Add some more domain specific features to the manual feature engineering process like:

  Days employed percentage, Income credit percentage, Payment rate, Payment Difference etc.
- Create Custom primitives for FeatureTools
- SMOTE to deal with class imbalance
- Code Clean up
- Wiki, README, ETHICS

References

- https://www.kaggle.com/willkoehrsen/start-here-a-gentle-introduction
- https://www.kaggle.com/codename007/home-credit-complete-eda-feature-importance
- https://docs.featuretools.com/index.html

Thank You

Questions???