EC48W FINAL REPORT

HOME CREDIT DEFAULT RISK

GROUP RANDOM FOREST:

Berk Filcan
Mehmet Akif Güçlü
Nihal Temüge
Beyza Yakıcı
Büşra Yavuz

Table of Contents

Abstract

Consumer credit scoring is often considered as a classification task where clients receive either a good or a bad credit status. In this paper, we have tried to develop a loan default prediction model by using dataset provided from the competition in Kaggle by Home Credit. We prioritized 96 features among 122 features. We dropped some columns due to missing values in our dataset. Then, we applied necessary data preparation steps such as missing variable treatment, correlation check and run the models. In order to select the best method, we applied three different machine learning methods which are Logistic regression, K-Nearest Neighbours Algorithm, Random Forest classifier. For our purposes, we compared these methods' scores with each other (namely; accuracy, recall, precision and f1 scores). We also look at the ROC-AUC and learning curves. According to the results, we select logistic regression (which is better to make predictions) for our research. Logistic regression has performed the best in most metrics both in training and test datasets. Results indicated that our logistic regression model is quite successful as we have 0.70 accuracy rate and additionally, our F1 score is 0.69, which is better than other models.

Introduction

A credit risk is usually defined as the risk of default on a debt that a borrower or other counterparty fails to meet their obligations to make required payments. There are significant examples of defaults in financial history, such as some parts of 1929 Wall Street Crash and especially 2009 Subprime Mortgage Crisis. For many banks around the world, loans are the main source of credit risk. To prevent potential defaults in payments, credit rating systems are employed by the banks and credit scores for borrowers are usually evaluated by institutions found by banks or independent financial institutions such as Moody's or S&D. Credit ratings are usually calculated by checking the borrower's payment history, the amount and the length of the debt and recent credits taken by the borrower. However, as scoring instruments imply, the assessment of risk requires a credit history and therefore, an assessment system is lacking for potential borrowers. The lack of such system creates struggles for people who have non-existent credit histories and makes them vulnerable to untrustworthy lenders, and increases the likelihood of moral hazard, causing banks to lend ineffectively. In our project, we will use the data taken from a Kaggle competition, provided by Home Credit Group, an international financial institution founded in Czechia, to explore different aspects and the likelihood of credit default risk. We will examine various characteristics of past borrowers to interpret future borrowers who has hardly any or non-existent credit histories.

Methods Review and Background

Our objective is to detect the customers who are likely to not pay their loans. For this purpose, we have selected Logistic Regression, Random Forest Classification, K-Nearest Neighbours Algorithm. In addition to these, we have created polynomials and we will take into consideration the Random Forest result for the polynomial, as well.

This is a standard supervised classification task. We call this supervised because the labels are included in the training data and the goal is to train a model to learn to predict the labels from the features. Also, this is a classification since the label is a binary variable, 0 (will repay loan on time), 1 (will have difficulty repaying loan).

For feature engineering purposes, we create polynomials and add them to the dataframe as columns. In polynomial features method, we make features that are powers of existing features as well as interaction terms between existing features. When we are creating polynomial features, we want to avoid using too high of a degree, both because the number of features scales exponentially with the degree, and because we can run into problems with overfitting. Additionally, we use domain knowledge features. We will make a couple features that attempt to capture what we think may be important for telling whether a client will default on a loan. These are as follows:

i. CREDIT_INCOME_PERCENT: the percentage of the credit amount relative to a client's income
ii. ANNUITY_INCOME_PERCENT: the percentage of the loan annuity relative to a client's income
iii. CREDIT_TERM: the length of the payment in months (since the annuity is the monthly amount due)
iv. DAYS_EMPLOYED_PERCENT: the percentage of the days employed relative to the client's age

Firstly, Logistic regression is the appropriate regression analysis to conduct when the dependent variable is dichotomous (binary). Like all regression analyses, the logistic regression is a predictive analysis. Logistic regression is used to describe data and to explain the relationship between one dependent binary variable and one or more nominal, ordinal, interval or ratio-level independent variables. In other words, the logistic regression model predicts P(Y=1) as a function of X. Our output that we test based on 1 or 0 value where 0 represents the people that are able to repay and 1 are those not able to pay.

Secondly, Random forest is a type of supervised machine learning algorithm based on ensemble learning in which you join different types of algorithms or same algorithm multiple times to form a more powerful prediction model. The random forest algorithm combines multiple algorithm of the same type i.e. multiple decision trees, resulting in a forest of trees, hence the name "Random Forest". We use this method for our main data and for the data we add the polynomials.

The KNN algorithm assumes that similar things exist in close proximity. In other words, similar things are near to each other.

We are going to use these 4 models and decide which one to use by looking at their performances. Our performance measurements are cross validation score, accuracy, precision, recall, F1 score, and also AUC (area under the curve).

We need to calculate True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN). A true positive is an outcome where the model correctly predicts the positive class. Similarly, a true negative is an outcome where the model correctly predicts the

negative class. A false positive is an outcome where the model incorrectly predicts the positive class. And a false negative is an outcome where the model incorrectly predicts the negative class. We are going to use these to calculate performance measurements and the confusion matrix. A confusion matrix is a table that is often used to describe the performance of a classification model on a set of test data for which the true values are known.

Cross-validation is a resampling procedure used to evaluate machine learning models on a limited data sample. The most common one is k-fold cross validation. The procedure is as follows:

1. Shuffle the dataset randomly.
2. Split the dataset into k groups
3. For each unique group:
    a. Take the group as a hold out or test data set
    b. Take the remaining groups as a training data set
    c. Fit a model on the training set and evaluate it on the test set
    d. Retain the evaluation score and discard the model
4. Summarize the skill of the model using the sample of model evaluation scores

Accuracy is the most intuitive performance measure and it is a ratio of correctly predicted observation to the total observations. Since the dataset we cleaned is balanced, this is a significant metric for us. Precision is the ratio of correctly predicted positive observations to the total predicted positive observations. High precision relates to the low false positive rate. Recall (sensitivity) is the ratio of correctly predicted positive observations to the all observations in actual class. F1 Score is the weighted average of Precision and Recall. Therefore, this score takes both false positives and false negatives into account. F1 score is better when there is a balanced matrix. These are calculated as follows:

- Accuracy = TP+TN/TP+FP+FN+TN
- Precision = TP/TP+FP
- Recall = TP/TP+FN
- F1 Score = 2*(Recall * Precision) / (Recall + Precision)

Data Cleaning and Restructuring

For our project we used the data provided by Home Credit Group in their Kaggle competition. Our objective is to forecast whether an applicant with a non-existent or negligible credit history will be able to make his/her payments based on historical loan application data. We had eight sources of data, namely; application_train, application_test, bureau, bureau_balance, previous_application, POS_CASH_BALANCE, credit_card_balance, installments_payment. First two datasets are our main focus since it contains information about loan and loan applicant at application time. 'TARGET' column, which contains binary values that provides information if the loans were paid or not, and other important variables derived from applicants are included in these datasets. Initially, we planned to use application_test dataset to test our results, but we decided that application_train dataset

was sufficient. Data can be accessed from https://www.kaggle.com/c/home-credit-default-risk/data.

```
float64    65
int64      41
object     16
dtype: int64
```

We had 307511 rows representing applicants, and 122 columns of total; 65 of which are floats, 41 of which are integers, and the rest is categorical. However, not all columns were ready to immediate use because they had significant amount of missing values. For example, COMMONAREA_MEDI column had 214865 missing values, which amounts to 69.9% of its rows. There were 67 columns with missing values, out of which 20 had a missing value percentage greater than 57%. The reason why we chose 57% as a threshold is not to exclude EXT_SOURCE_1, which had a high correlation with TARGET variable. To examine our data further, we dropped the columns over this threshold. Also, FLAG_MOBIL and FLAG_DOCUMENT_2 columns had a single value all over the column, so they were also dropped since they did not have any significance. Still, there were rows with missing values, so, data cleaning and restructuring was needed when it comes to build our machine learning models. Therefore, we had two options: either imputation or removing rows. When we dropped these columns, we still had sufficient number of applicants, thus, imputation was not necessary. After carrying out these changes, the dimensions of our dataset were reduced to 36264x98.

The values of DAYS_BIRTH and DAYS_EMPLOYED had negative values because they had been calculated by defining the application date as reference point; therefore, we took their absolute values for the sake of calculations. SK_ID_CURR column, ID numbers that are assigned to applicants, were dropped since they were unnecessary for analysis. For categorical values, such as CODE_GENDER and NAME_EDUCATION_TYPE, we encoded the data. If a categorical column had 2 or fewer unique categories, its values are directly assigned as 0 or 1, i.e. integer encoding is used. If a categorical column had more than 2 unique categories, integer encoding is not sufficient since there is not any ordinal relationship in unique categories. Such columns were encoded using one-hot encoding method.

To decide which data might be useful for machine learning purposes, we ran correlation tests to see which ones are correlated with TARGET variable. As a result of the test, we employed 12 columns out of the remedial 98. TARGET (whether the loan is paid), NAME_EDUCATION_TYPE (applicant's degree), OCCUPATION_TYPE (applicant's job class), CODE_GENDER (applicant's gender), NAME_INCOME_TYPE (applicant's current working status), DAYS_BIRTH (how many days passed since the applicant has born), DAYS_EMPLOYED (number of days of an applicant is employed), CNT_CHILDREN (number of an applicant's children), AMT_INCOME_TOTAL (total income of applicant), and three other external variables, provided by Home Credit Group, namely EXT_SOURCE_1, EXT_SOURCE_2, EXT_SOURCE_3. As expected, CNT_CHILDREN and CNT_FAMILY_MEMBERS are highly correlated. Also, we found that the correlation between DAYS_BIRTH and EXT_SOURCE_1 is significant which might imply that the latter is calculated using the former. We will use this smaller dataframe while looking at the heatmap.

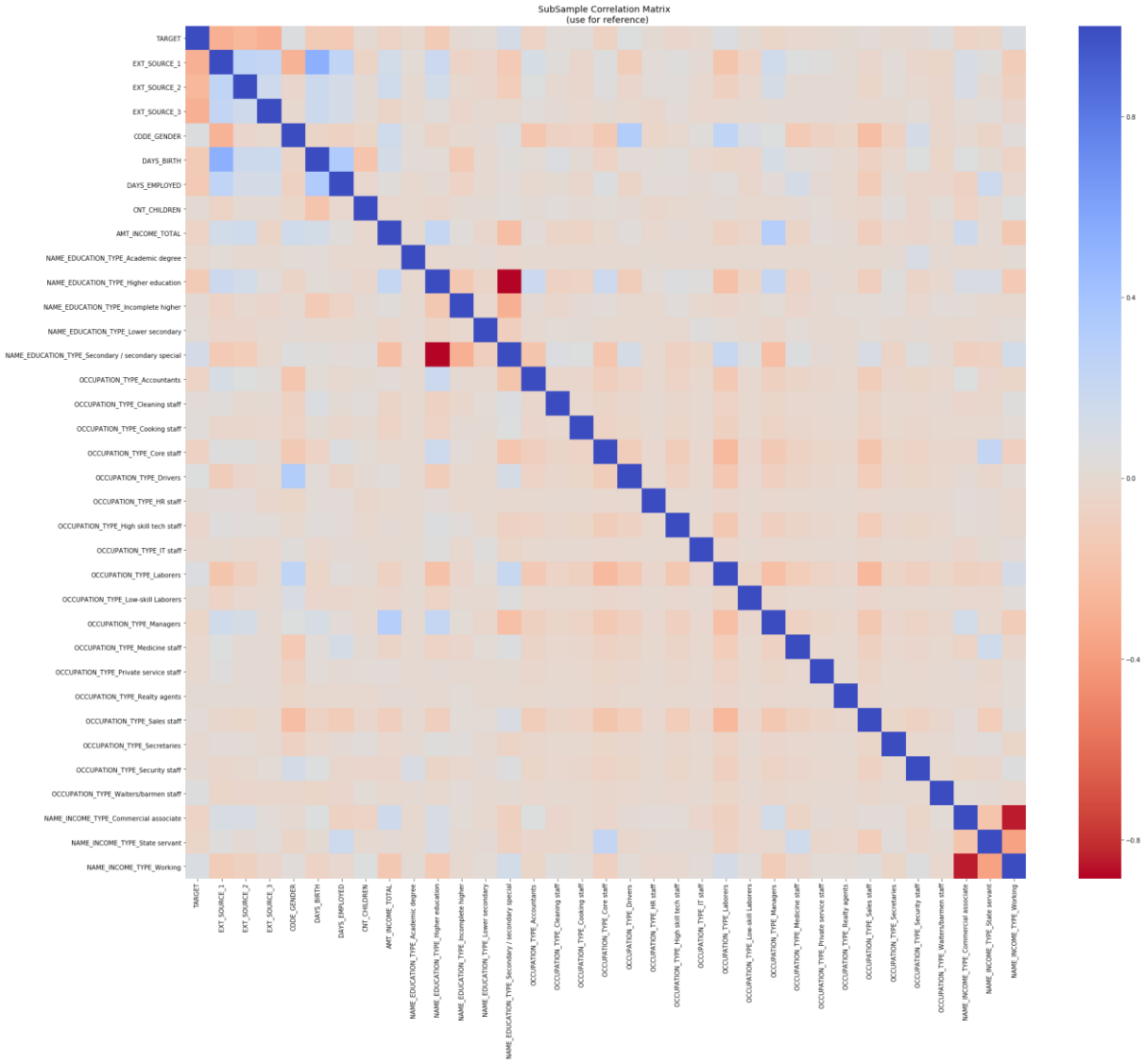
After we cleaned and restructured our dataset, we tried to draw its general picture. The classes were heavily skewed in TARGET variable column. The percentage of value 0, paid loans, was 93.43. To examine our data properly and define its aspects clearly, we created a subsample dataset which has an equal distribution of TARGET values, 1 and 0. Values of 0 were randomly chosen. We frequently use this dataset throughout our analysis, but an oversampled dataset created with SMOTE method is also employed.

Statistical Analysis and Data Exploration

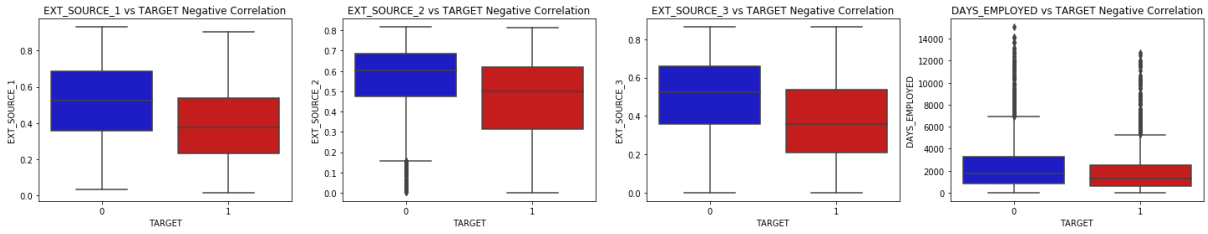
As stated in the previous part, we created a smaller dataframe consisting 'TARGET', 'EXT_SOURCE_1', 'EXT_SOURCE_2', 'EXT_SOURCE_3', 'NAME_EDUCATION_TYPE', 'OCCUPATION_TYPE', 'CODE_GENDER', 'NAME_INCOME_TYPE', 'DAYS_BIRTH', 'DAYS_EMPLOYED', 'CNT_CHILDREN', 'AMT_INCOME_TOTAL'. Below, we see the main statistics for these before encoding.

	TARGET	EXT_SOURCE_1	EXT_SOURCE_2	EXT_SOURCE_3	DAYS_BIRTH	DAYS_EMPLOYED	CNT_CHILDREN	AMT_INCOME_TOTAL
count	36264.000000	36264.000000	36264.000000	36264.000000	36264.000000	36264.000000	36264.000000	3.626400e+04
mean	0.065685	0.506168	0.549828	0.494356	14299.107048	2368.465007	0.539902	1.936332e+05
std	0.247734	0.205886	0.174524	0.194336	3366.730682	2259.202504	0.755866	1.145574e+05
min	0.000000	0.014691	0.000010	0.000527	7680.000000	4.000000	0.000000	2.700000e+04
25%	0.000000	0.344601	0.458488	0.352340	11628.000000	780.000000	0.000000	1.260000e+05
50%	0.000000	0.510588	0.596036	0.513694	14004.000000	1669.000000	0.000000	1.710000e+05
75%	0.000000	0.672684	0.680748	0.651260	16627.000000	3189.000000	1.000000	2.250000e+05
max	1.000000	0.941986	0.855000	0.887664	25017.000000	16495.000000	19.000000	4.500000e+06

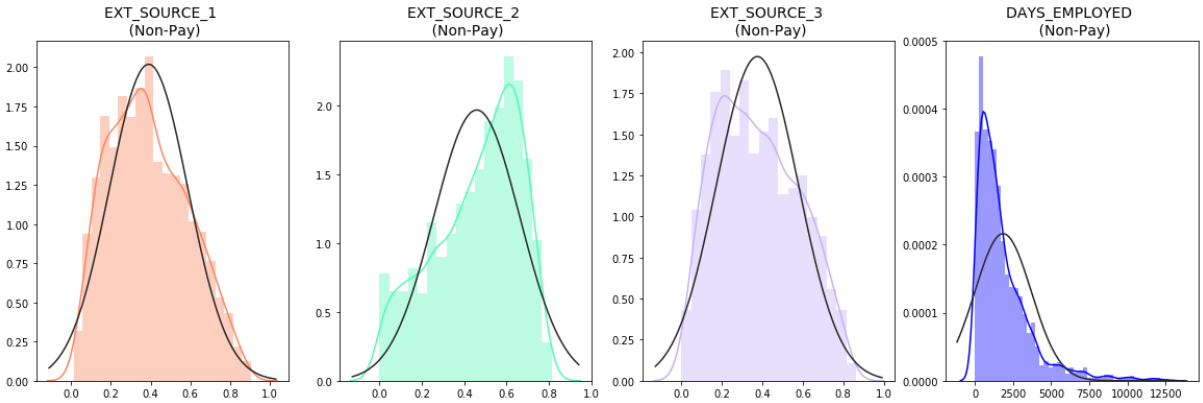
In order to explore our sub-sampled data more deeply, we first construct a correlation matrix between the features. A correlation matrix is simply a table which displays the correlation coefficients for different variables. The matrix depicts the correlation between all the possible pairs of values in a table. It is a powerful tool to summarize a large dataset and to identify and visualize patterns in the given data. In our case, it is important that we use the correct subsample in order for us to see which features have a high positive or negative correlation with regards to TARGET. It can be seen that TARGET is highly correlated with EXT_SOURCE_1, EXT_SOURCE_2, EXT_SOURCE_3 and DAYS_EMPLOYED. So, we use boxplots for having a better understanding of the distribution of these features in default and non-default. The lower the values of these features are, the more likely the end result will be pay, in other words it'll be a non-default since they are negatively correlated.



When we look at the boxplots, we see that in our EXT_SOURCE_1, EXT_SOURCE_2 EXT_SOURCE_3 data, there are no extreme outliers so much and we decided to not remove them. DAYS_EMPLOYED has some extreme outliers. When we remove them, we saw that it did not change our original data and results. So, we kept them, as well.



In addition to this, we drew EXT_SOURCE_1, EXT_SOURCE_2, EXT_SOURCE_3 and DAYS_EMPLOYED's distribution of non-payer customers and we received same results. EXT_SOURCE_1, EXT_SOURCE_2 and EXT_SOURCE_3 are close to normal distribution and have not a lot of extreme outliers. However, there are some extreme outliers in DAYS_EMPLOYED data.



Methods and Tools of Use

We use Logistic Regression, KNN algorithm, and Random Forest. We also add polynomials and use it in Random Forest model. We have compared these 4 models according to their cross-validation score, learning curve, ROC-AUC, confusion matrix, accuracy, precision, recall and f1 score. Besides, we also consider the learning curve. We choose the model that gives better scores among those criteria.

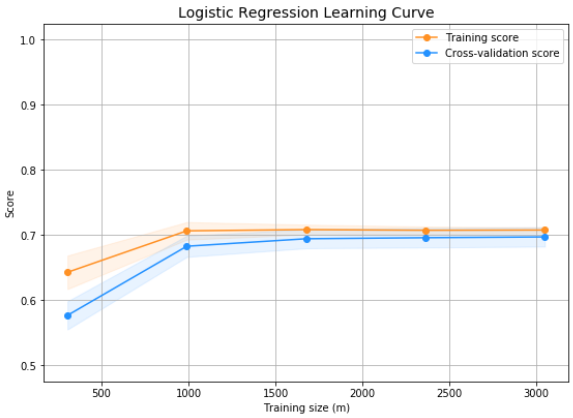
First, we look at the cross validation score below. GridSearchCV is used to determine the parameters that gives the best predictive score for the classifiers. The wider the gap between the standard cross-validation and the cross-validation score with GridSearchCV, the more likely the model is overfitting (high variance). If the score is low in both cross-validation sets this is an indication that our model is underfitting (high bias). We see the results for both the standard score and the score by using best estimators below, respectively:

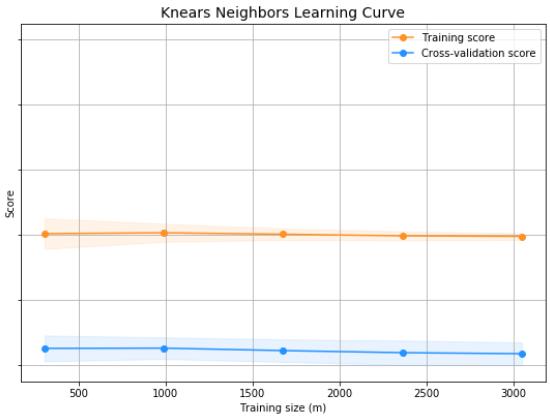
```
LogisticRegression  Cross Validation Score: 56.99999999999999 %
KNeighborsClassifier  Cross Validation Score: 53.0 %
RandomForestClassifier  Cross Validation Score: 63.0 %
RandomForestClassifier on Polynomial Feature  Cross Validation Score: 66.0 %
```

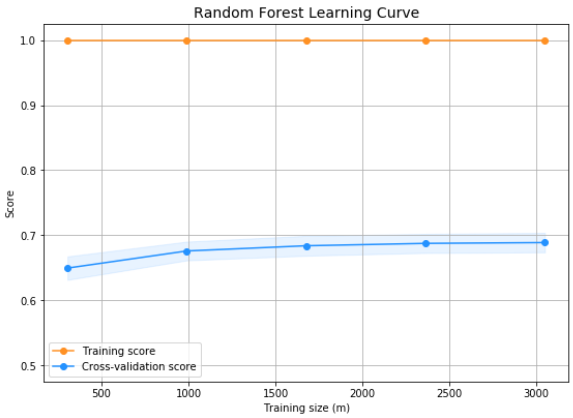
```
Logistic Regression Cross Validation Score:  69.88%
Knears Neighbors Cross Validation Score 52.35%
Random Forest Cross Validation Score: 68.64%
Random Forest Poly Cross Validation Score: 69.14%
```

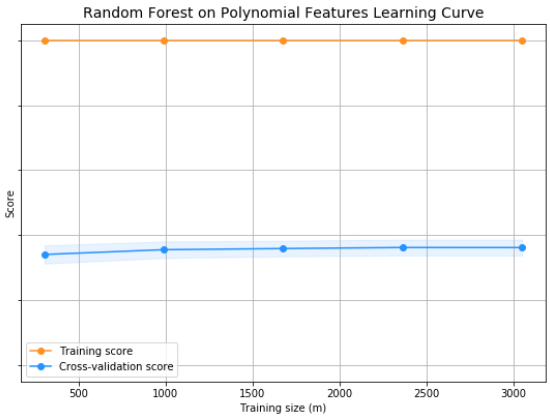
Among all these models, KNN has the worst performance at this category. Logistic Regression and Random Forest with the polynomial features give better results, yet we should remember that we may be facing an overfitting problem for the latter. We will look at other metrics to get a more precise result.

Another comparison method is to look at their learning curves. As we see in the following figure, Logistic Regression has a better learning curve than other models.



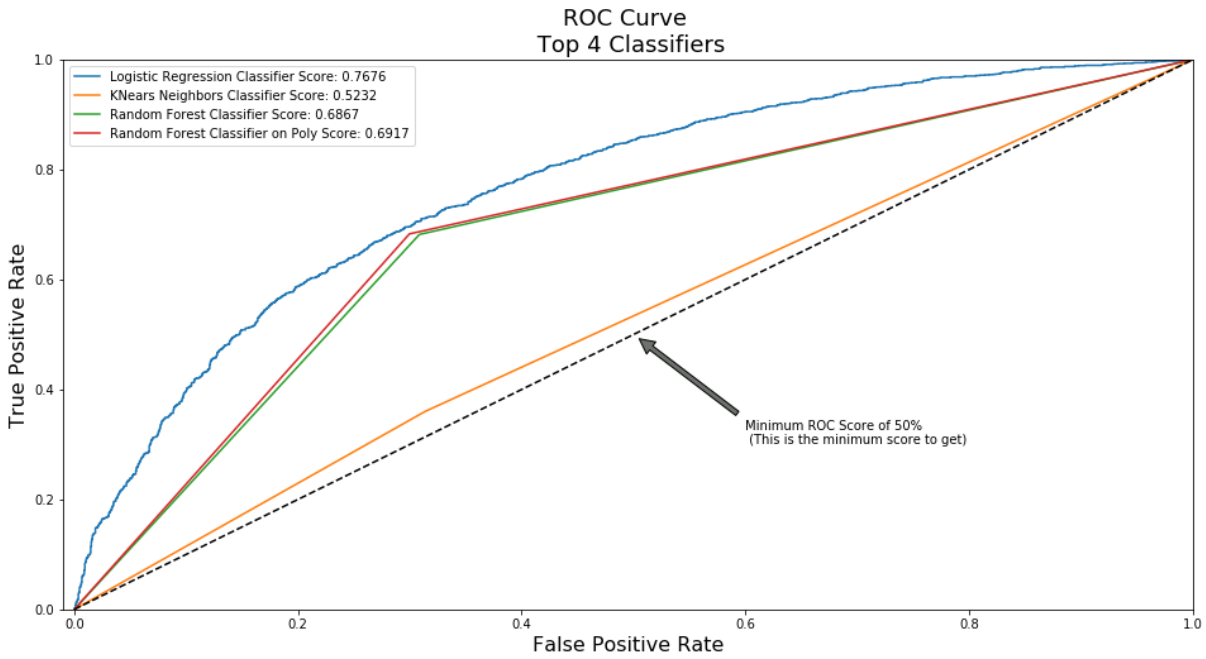




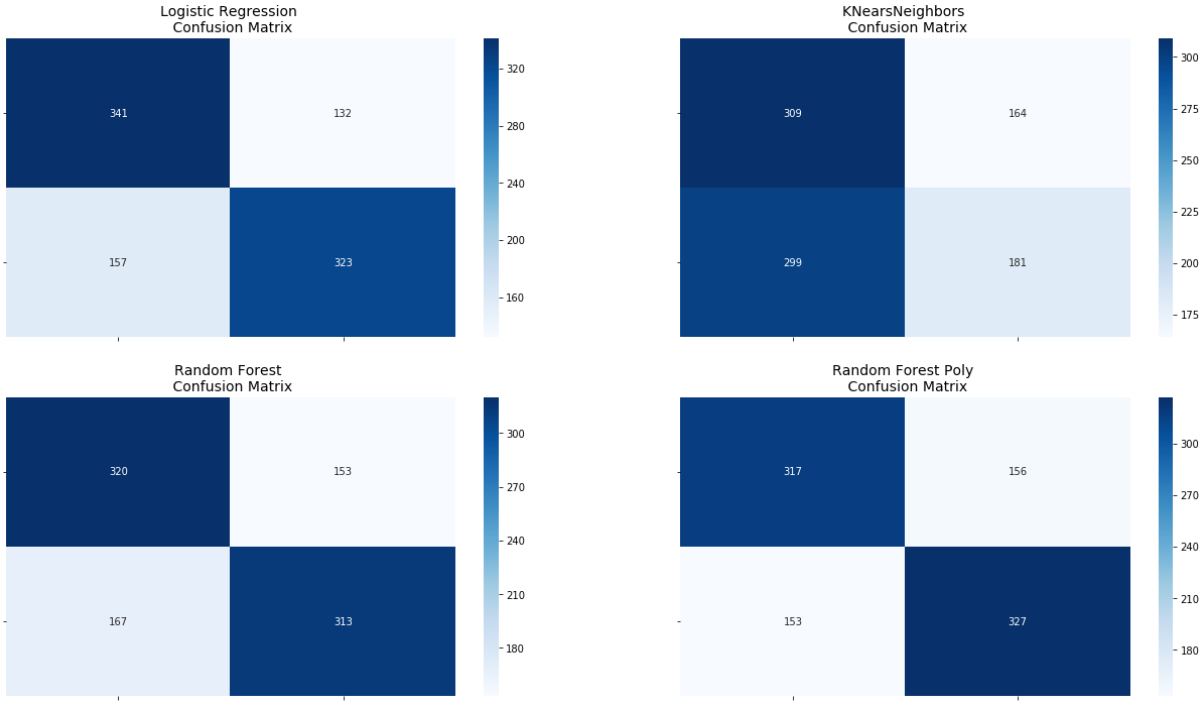


We also look at the ROC-AUC in order to measure the performance of the models. Logistic Regression has the best Receiving Operating Characteristic score (ROC), meaning that Logistic Regression quite accurately separates pay and non-pay structure. The closer the curve to the diagonal the worse the model since it will be close to a random model. Since area under the curve is the highest in the Logistic Regression, that model is clearly better. We also see that the model that we have included polynomials performs slightly better than the main Random Forest. However, it is still lower than Logistic Regression's. KNN performs the worst, again. AUC results and the ROC curve is stated below:

```
Logistic Regression AUC:  0.7676323177774875
KNears Neighbors AUC:  0.5231851834715078
Random Forest AUC:  0.6866876365701456
Random Forest Poly AUC:  0.6916659919061791
```



In addition to these, we will look at the confusion matrix. Also, using the values below, we will calculate accuracy, precision, recall, and F1 score.



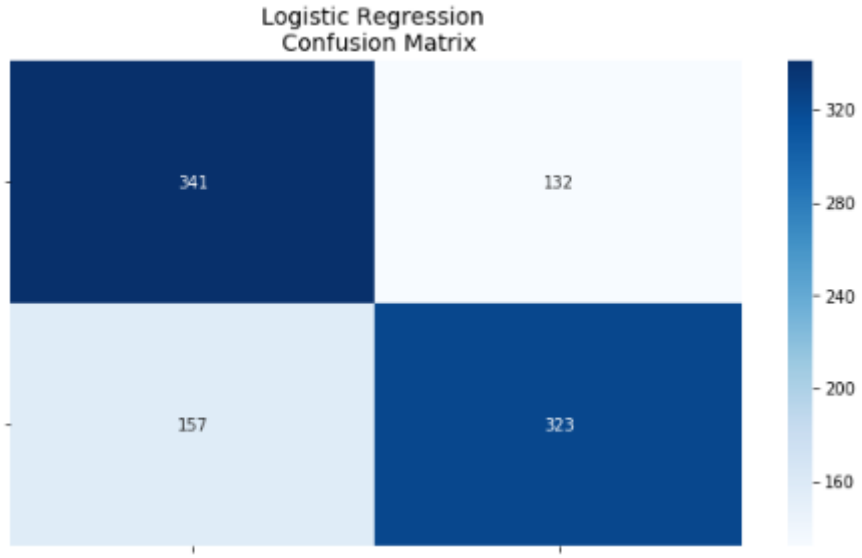
Lastly, in the below table, we see that Logistic Regression performs better than KNN and Random Forest in all aspects.

```
Logistic Regression:
              precision    recall  f1-score   support

           0       0.68      0.72      0.70       473
           1       0.71      0.67      0.69       480

    accuracy                           0.70       953
   macro avg       0.70      0.70      0.70       953
weighted avg       0.70      0.70      0.70       953

KNears Neighbors:
              precision    recall  f1-score   support

           0       0.51      0.65      0.57       473
           1       0.52      0.38      0.44       480

    accuracy                           0.51       953
   macro avg       0.52      0.52      0.51       953
weighted avg       0.52      0.51      0.50       953

Random Forest Neighbors:
              precision    recall  f1-score   support

           0       0.66      0.68      0.67       473
           1       0.67      0.65      0.66       480

    accuracy                           0.66       953
   macro avg       0.66      0.66      0.66       953
weighted avg       0.66      0.66      0.66       953

Random Forest Poly Neighbors:
              precision    recall  f1-score   support

           0       0.67      0.67      0.67       473
           1       0.68      0.68      0.68       480

    accuracy                           0.68       953
   macro avg       0.68      0.68      0.68       953
weighted avg       0.68      0.68      0.68       953
```

To sum up, Logistic Regression classifier is more accurate than the other three classifiers in most cases. For this reason, we select this model in our machine learning algorithm.

Results

We calculated the performance of our logistic regression model, which is trained with both random under-sampled and oversampled (SMOTE technique) and 80-20 split data, the highest between all models.



True Negatives (Top-Left Square): This is the number of **correctly** classifications of the "Payer" class.

False Negatives (Top-Right Square): This is the number of **incorrectly** classifications of the "Payer class.

False Positives (Bottom-Left Square): This is the number of **incorrectly** classifications of the "Non-Payer" class

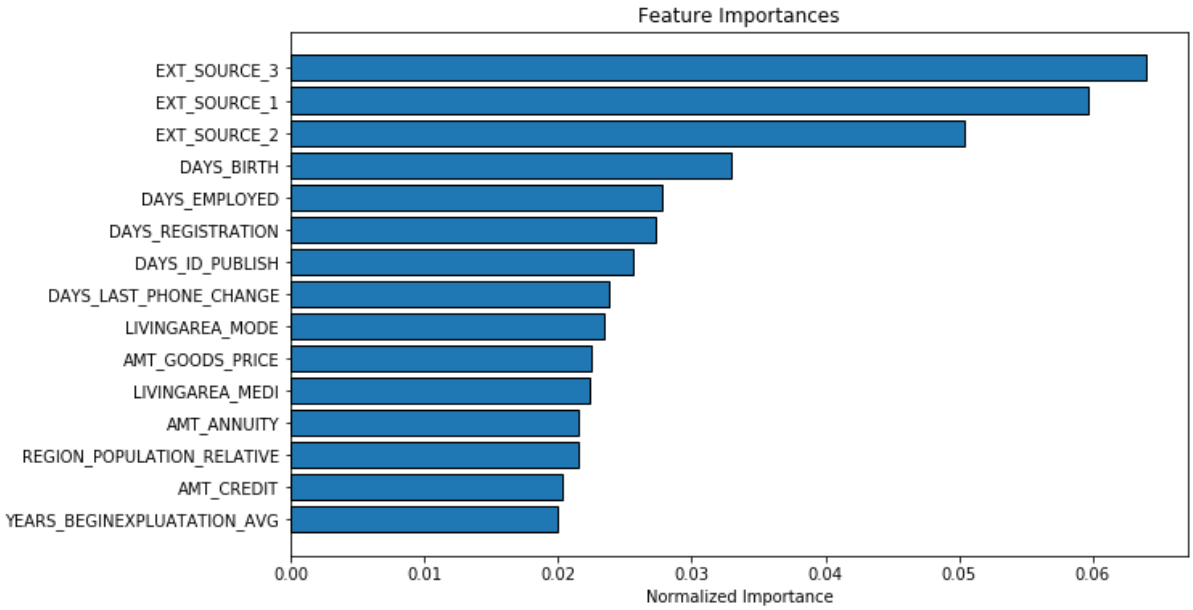
True Positives (Bottom-Right Square): This is the number of **correctly** classifications of the "Non-Payer" class.

	Technique	Score
0	Random UnderSampling	0.696747
1	Oversampling (SMOTE)	0.712217

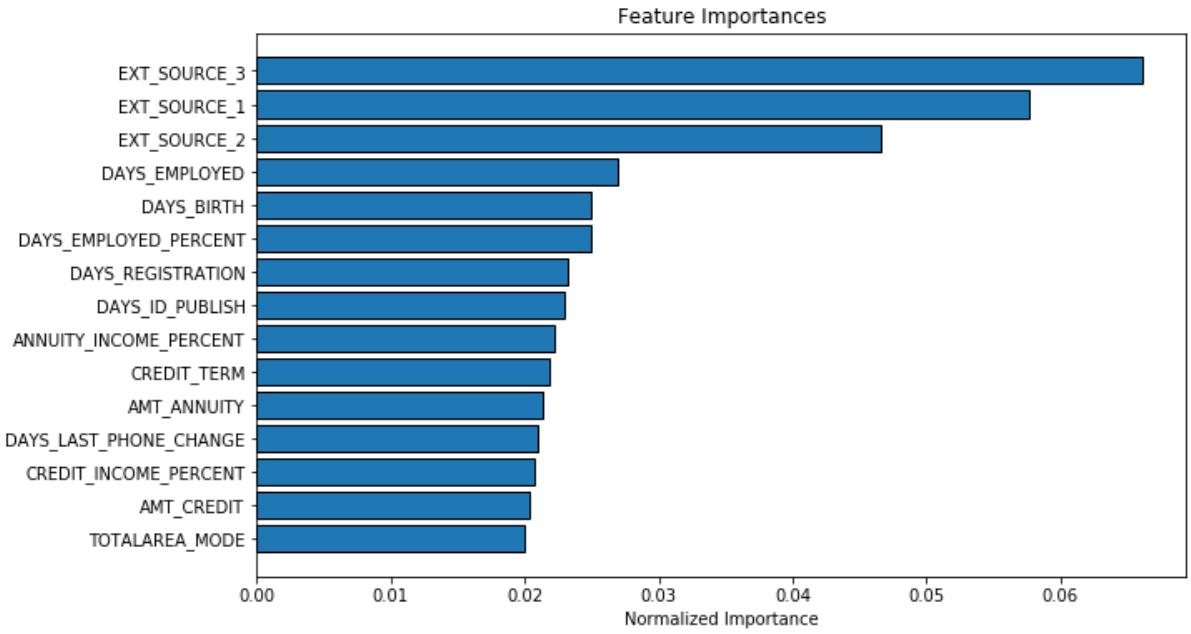
Our accuracy is quite high with 0.70 with Random Undersampling and 0.71 with Oversampling (SMOTE). These scores could have been improved with more data. Our data was imbalanced, hence we had to create our data with small amount of the original data. Moreover, some of the features had more than 70% missing values. That was another reason to decrease the data size. It is a well-known fact that larger sample sizes provide more accurate mean values, identify outliers that could skew the data in a smaller sample and provide a smaller margin of error. These results are better than most of Kaggle contest competitors' results.

To look at which variables are the most relevant, we can look at the feature importances of the random forest algorithm. Given the correlations, we saw in the exploratory data analysis,

we should expect that the most important features are the EXT_SOURCEs and the DAYS_EMPLOYED.



As expected, the most important features are those dealing with EXT_SOURCE and DAYS_EMPLOYED. We can see that there are only small parts of the features that are important for our model. Dropping some features from our dataframe might not decrease our model's prediction performance very much. Feature importance is an easy and useful method to use in dimensionality reduction. Moreover, it helps us to understand which factors are more important than others when making a prediction.



Additionally, we see that all four of our hand-engineered features made it into the top 15 most important! This should give us confidence that our domain knowledge was at least partially on track.

Conclusion

Credit scoring has a key role in credit risk management. Machine learning approaches can be easily used for the consistent estimation of individual consumer credit risks. We believe that the flexibility of machine learning models can provide strong support in credit scoring dealing with rich and complicated information in credit risk assessment. In this study, we employ a unique, very large dataset (which consists of anonymous information) from the competition in Kaggle by Home Credit in order to build and test Logistic regression, K-Nearest Neighbour, Random Forest classifier for predicting credit card risk. We continued our research by trying to find the best fitting method for this prediction. When applied to a large credit scoring data set, the test results clearly indicate that logistic regression outperforms other models. The lowest accuracy and F1 score belongs to the K-Nearest Neighbours method and the winner of this comparison is the Logistic Regression with the 0.70 accuracy score and 0.69 F1 score (the highest scores).

Resources and References

"Bank Loan Default Prediction." Kaggle. Accessed July 16, 2019. https://www.kaggle.com/zhunqiang/bank-loan-default-prediction?scriptVersionId=7634496.

Bozkurt, Gul Efsan. Feature Selection and Transfer Learning Algorithms with Applications on Credit Risk Analysis. Master's thesis, Bogazici University, 2012. Accessed July 31, 2019. https://en.academicresearch.net/feature-selection-and-transfer-learning-algorithms-with-applications-on-credit-risk-analysis/.

Butaru, Florentin, Qingqing Chen, Brian Clark, Sanmay Das, Andrew Lo, and Akhtar Siddique. "Risk and Risk Management in the Credit Card Industry." 2015.

"Credit Fraud || Dealing with Imbalanced Datasets." Kaggle. Accessed July 20, 2019. https://www.kaggle.com/janiobachmann/credit-fraud-dealing-with-imbalanced-datasets.

Donnelly, Catherine, and Paul Embrechts. "The Devil Is in the Tails: Actuarial Mathematics and the Subprime Mortgage Crisis." ASTIN Bulletin40, no. 1 (2010).

Eichengreen, Barry, and Kris James Mitchener. "The Great Depression as a Credit Boom Gone Wrong." SSRN Electronic Journal, 2003.

"Home Credit Default Risk." Kaggle. Accessed June 25, 2019. https://www.kaggle.com/c/home-credit-default-risk/data.

James, Gareth. An Introduction to Statistical Learning: With Applications in R. New York: Springer, 2017.

Kruppa, Jochen, Alexandra Schwarz, Gerhard Arminger, and Andreas Ziegler. "Consumer Credit Risk: Individual Probability Estimates Using Machine Learning." Expert Systems with Applications40, no. 13 (2013).

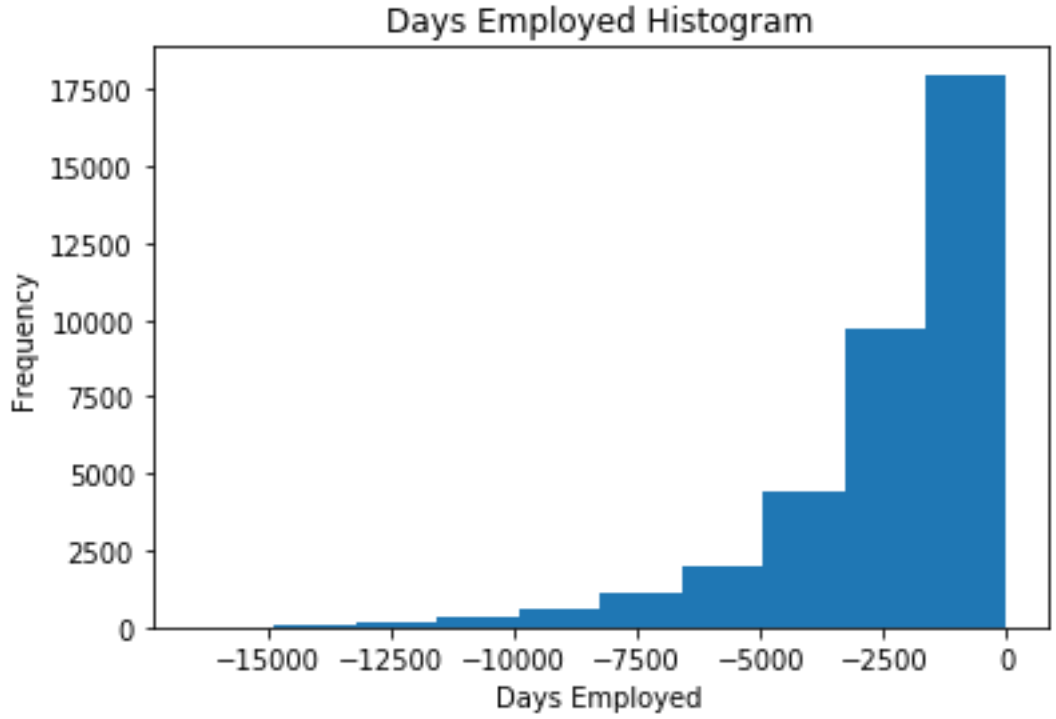
Lantz, Brett. Machine Learning with R: Discover How to Build Machine Learning Algorithms, Prepare Data, and Dig Deep into Data Prediction Techniques with R. Birmingham: Packt Publishing, 2013.

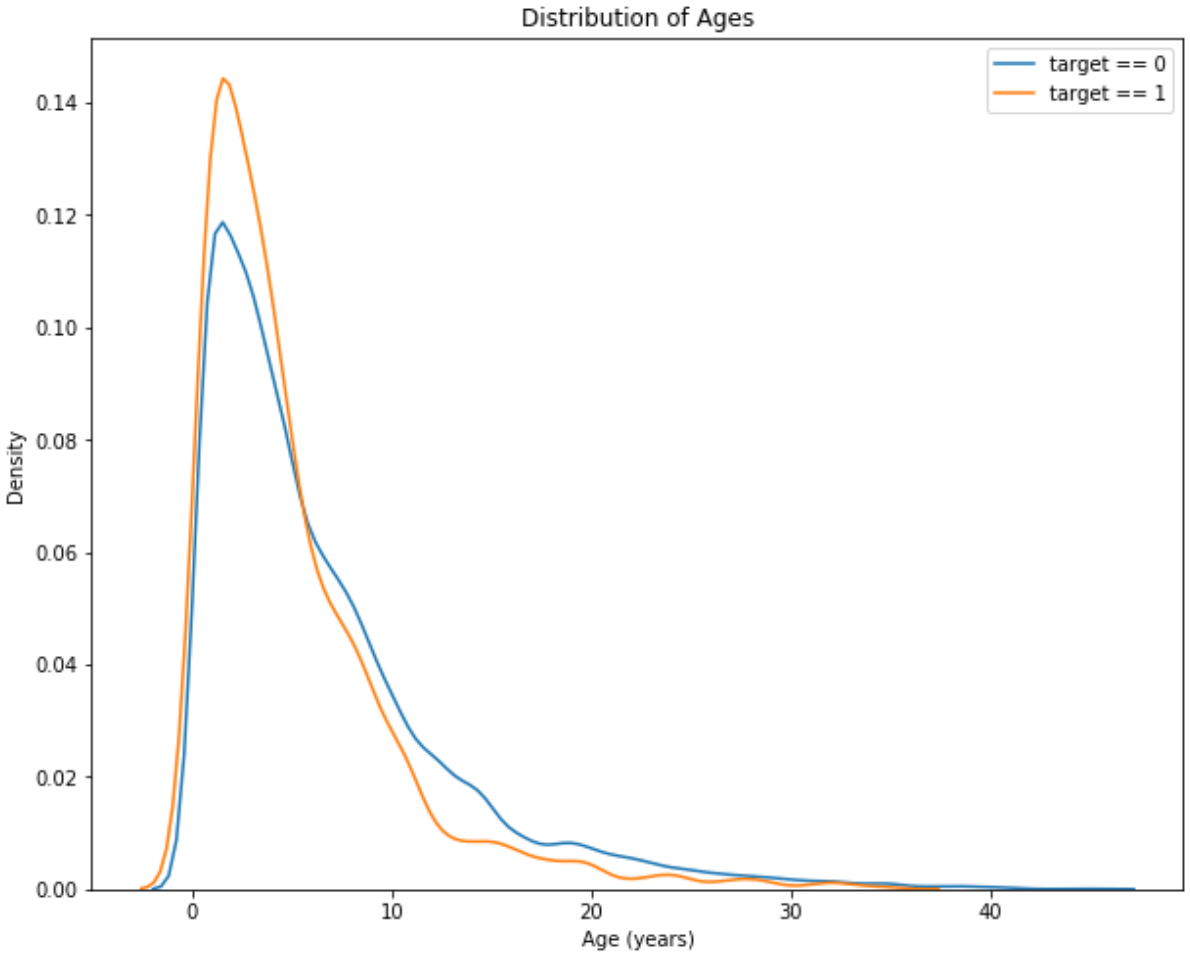
Luo, Cuicui, Desheng Wu, and Dexiang Wu. "A Deep Learning Approach for Credit Scoring Using Credit Default Swaps." Engineering Applications of Artificial Intelligence65 (December 2016).

Sengupta, Arindam. "A First Course in Machine Learning by Simon Rogers and Mark Girolami." International Statistical Review82, no. 1 (2014).

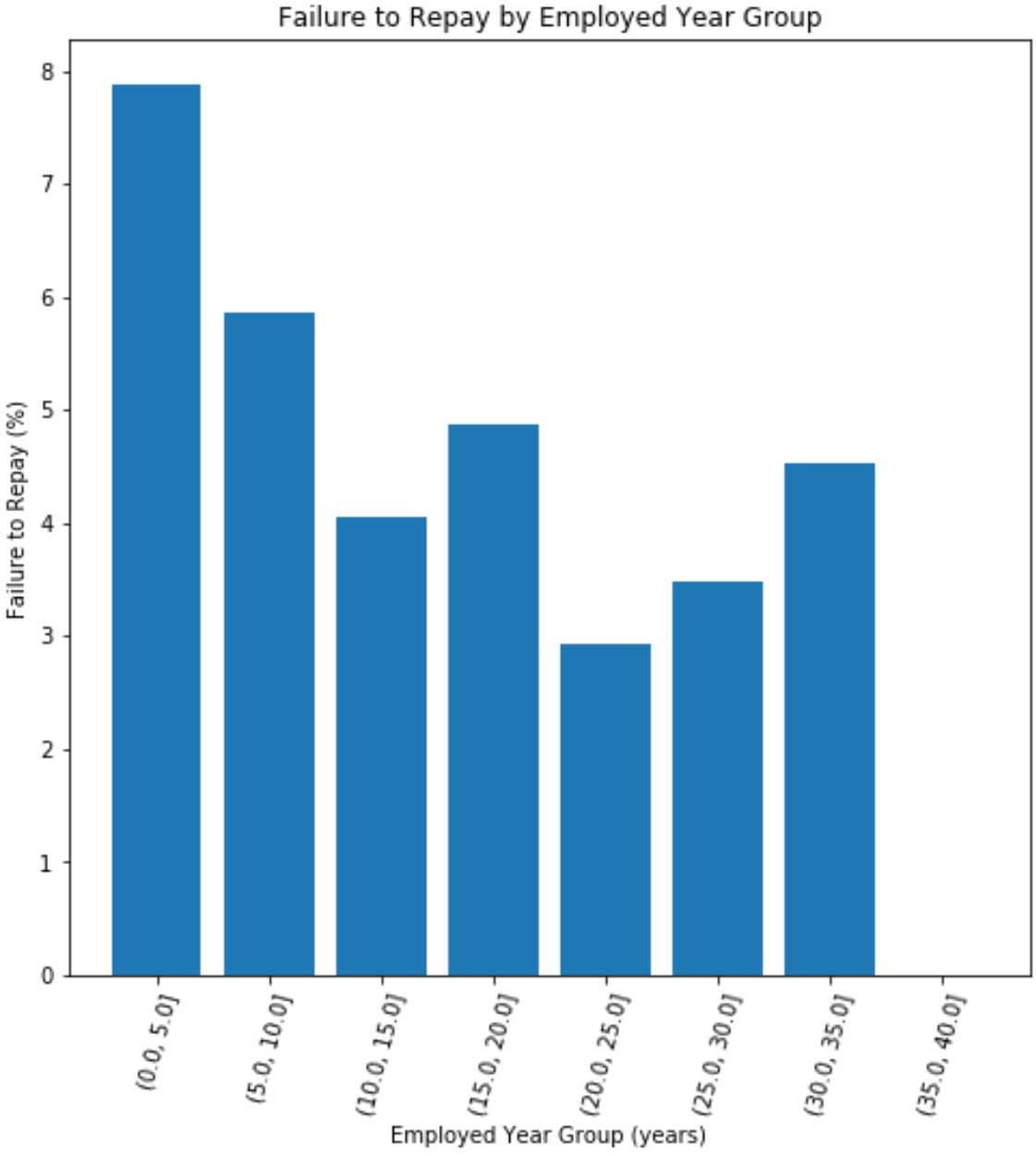
"Start Here: A Gentle Introduction." Kaggle. Accessed June 25, 2019. https://www.kaggle.com/willkoehrsen/start-here-a-gentle-introduction.

Appendix


Days Employed Histogram
Frequency
17500
15000
12500
10000
7500
5000
2500
0
−15000
−12500
−10000
−7500
−5000
−2500
0
Days Employed


Distribution of Ages
target == 0
target == 1
Density
0.14
0.12
0.10
0.08
0.06
0.04
0.02
0.00
0
10
20
30
40
Age (years)

	TARGET	DAYS_EMPLOYED	YEARS_BINNED
0	1	1.745205	(0.0, 5.0]
25	0	9.572603	(5.0, 10.0]
50	0	3.221918	(0.0, 5.0]
51	0	19.115068	(15.0, 20.0]
55	0	11.249315	(10.0, 15.0]
57	0	6.441096	(5.0, 10.0]
65	0	1.586301	(0.0, 5.0]
70	0	7.791781	(5.0, 10.0]
71	0	2.443836	(0.0, 5.0]
93	0	3.421918	(0.0, 5.0]

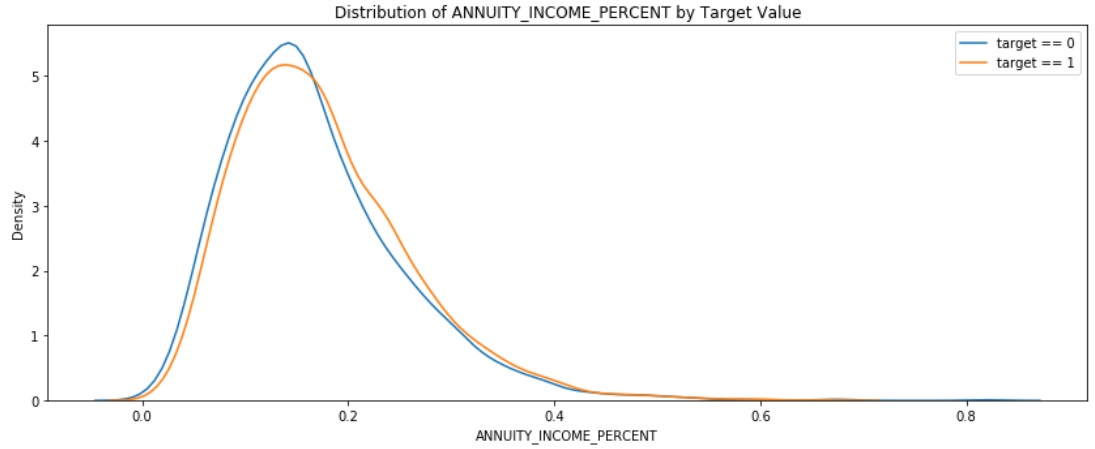


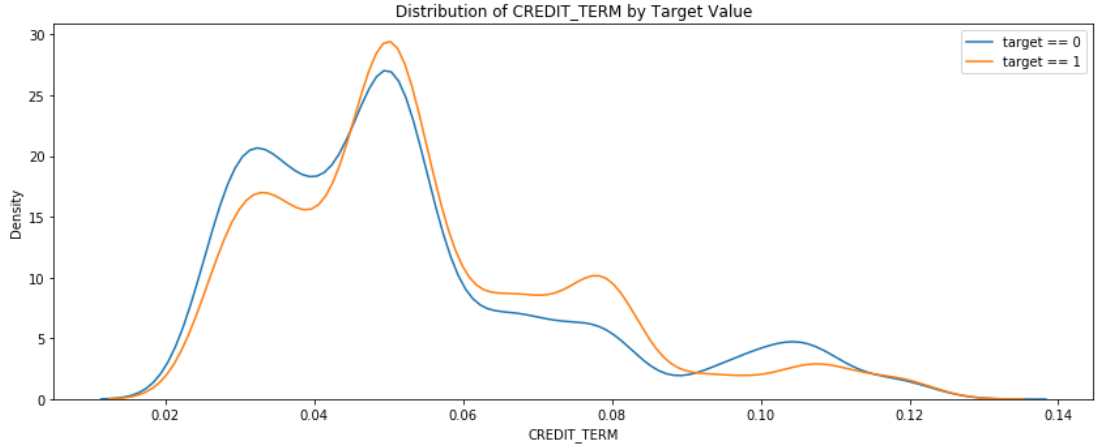
FEATURE LIST

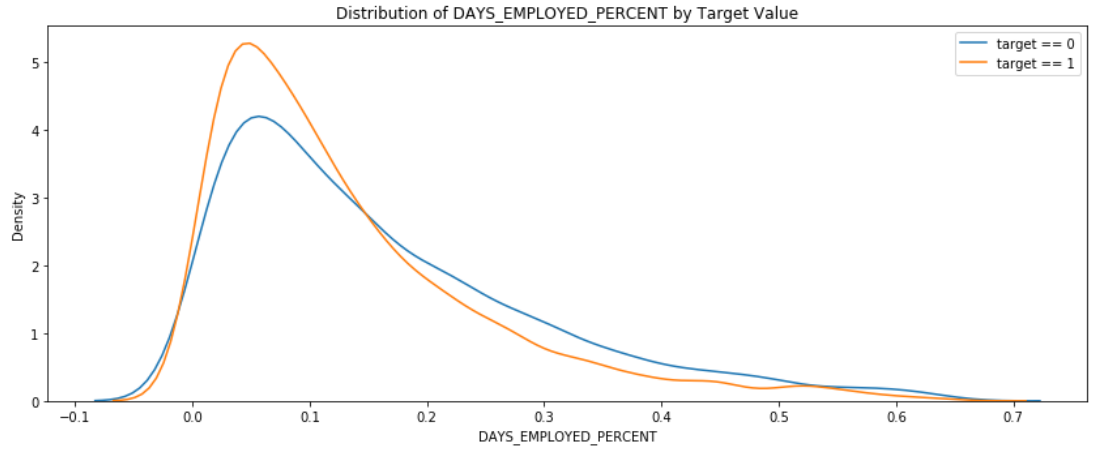
```
['TARGET',
 'NAME_CONTRACT_TYPE',
 'CODE_GENDER',
 'FLAG_OWN_CAR',
 'FLAG_OWN_REALTY',
 'CNT_CHILDREN',
 'AMT_INCOME_TOTAL',
 'AMT_CREDIT',
 'AMT_ANNUITY',
 'AMT_GOODS_PRICE',
 'NAME_TYPE_SUITE',
 'NAME_INCOME_TYPE',
 'NAME_EDUCATION_TYPE',
 'NAME_FAMILY_STATUS',
 'NAME_HOUSING_TYPE',
 'REGION_POPULATION_RELATIVE',
 'DAYS_BIRTH',
 'DAYS_EMPLOYED',
 'DAYS_REGISTRATION',
 'DAYS_ID_PUBLISH',
 'FLAG_EMP_PHONE',
 'FLAG_WORK_PHONE',
 'FLAG_CONT_MOBILE',
 'FLAG_PHONE',
 'FLAG_EMAIL',
 'OCCUPATION_TYPE',
 'CNT_FAM_MEMBERS',
 'REGION_RATING_CLIENT',
 'REGION_RATING_CLIENT_W_CITY',
 'WEEKDAY_APPR_PROCESS_START',
 'HOUR_APPR_PROCESS_START',
 'REG_REGION_NOT_LIVE_REGION',
 'REG_REGION_NOT_WORK_REGION',
 'LIVE_REGION_NOT_WORK_REGION',
 'REG_CITY_NOT_LIVE_CITY',
 'REG_CITY_NOT_WORK_CITY',
 'LIVE_CITY_NOT_WORK_CITY',
 'ORGANIZATION_TYPE',
 'EXT_SOURCE_1',
 'EXT_SOURCE_2',
 'EXT_SOURCE_3',
 'APARTMENTS_AVG',
 'YEARS_BEGINEXPLUATATION_AVG',
 'ELEVATORS_AVG',
 'ENTRANCES_AVG',
 'FLOORSMAX_AVG',
 'LIVINGAREA_AVG',
 'NONLIVINGAREA_AVG',
 'APARTMENTS_MODE',
 'YEARS_BEGINEXPLUATATION_MODE',
 'ELEVATORS_MODE',
 'ENTRANCES_MODE',
 'FLOORSMAX_MODE',
 'LIVINGAREA_MODE',
 'NONLIVINGAREA_MODE',
 'APARTMENTS_MEDI',
```

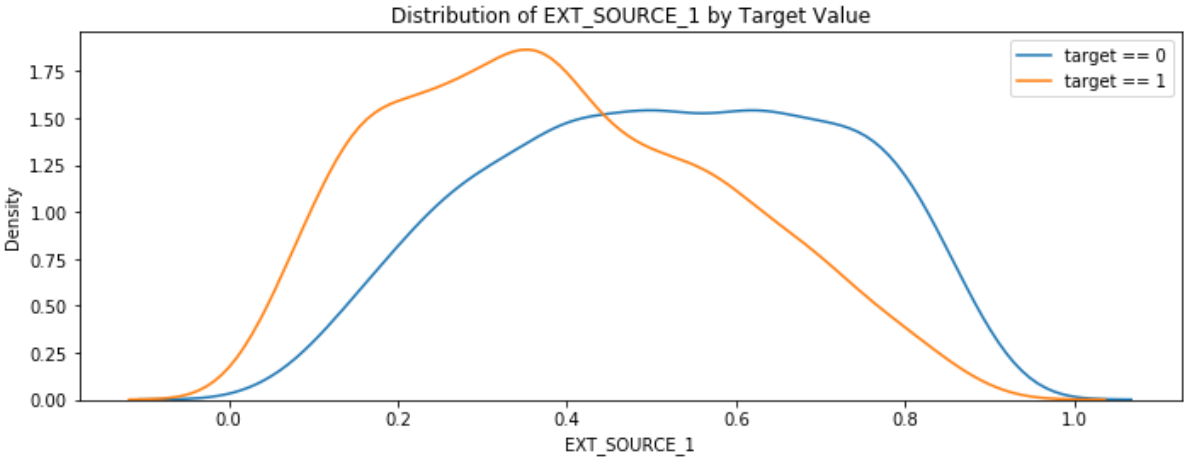
```
'YEARS_BEGINEXPLUATATION_MEDI',
'ELEVATORS_MEDI',
'ENTRANCES_MEDI',
'FLOORSMAX_MEDI',
'LIVINGAREA_MEDI',
'NONLIVINGAREA_MEDI',
'HOUSETYPE_MODE',
'TOTALAREA_MODE',
'WALLSMATERIAL_MODE',
'EMERGENCYSTATE_MODE',
'OBS_30_CNT_SOCIAL_CIRCLE',
'DEF_30_CNT_SOCIAL_CIRCLE',
'OBS_60_CNT_SOCIAL_CIRCLE',
'DEF_60_CNT_SOCIAL_CIRCLE',
'DAYS_LAST_PHONE_CHANGE',
'FLAG_DOCUMENT_3',
'FLAG_DOCUMENT_4',
'FLAG_DOCUMENT_5',
'FLAG_DOCUMENT_6',
'FLAG_DOCUMENT_7',
'FLAG_DOCUMENT_8',
'FLAG_DOCUMENT_9',
'FLAG_DOCUMENT_10',
'FLAG_DOCUMENT_11',
'FLAG_DOCUMENT_12',
'FLAG_DOCUMENT_13',
'FLAG_DOCUMENT_14',
'FLAG_DOCUMENT_15',
'FLAG_DOCUMENT_16',
'FLAG_DOCUMENT_17',
'FLAG_DOCUMENT_18',
'FLAG_DOCUMENT_19',
'FLAG_DOCUMENT_20',
'FLAG_DOCUMENT_21',
'AMT_REQ_CREDIT_BUREAU_HOUR',
'AMT_REQ_CREDIT_BUREAU_DAY',
'AMT_REQ_CREDIT_BUREAU_WEEK',
'AMT_REQ_CREDIT_BUREAU_MON',
'AMT_REQ_CREDIT_BUREAU_QRT',
'AMT_REQ_CREDIT_BUREAU_YEAR']
```

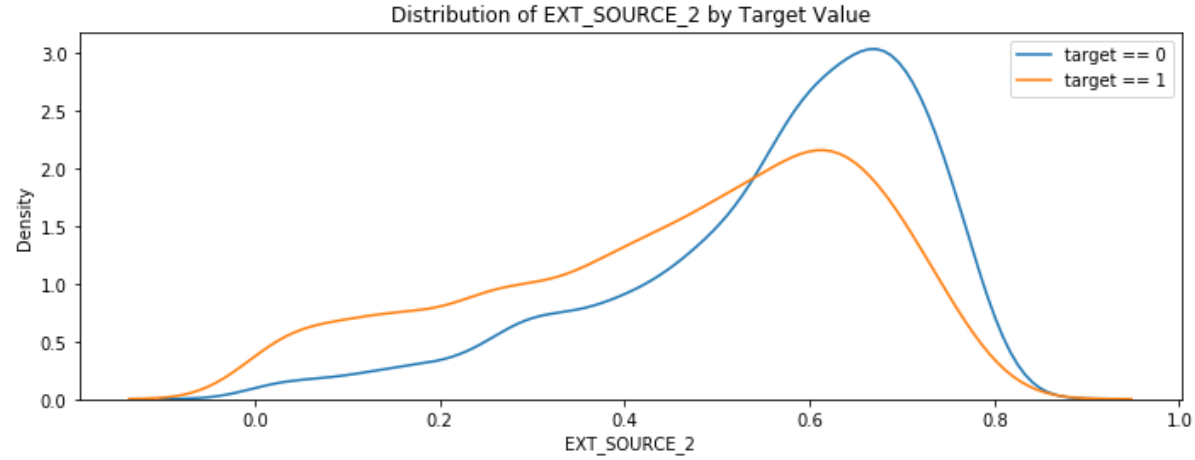

Distribution of CREDIT_INCOME_PERCENT by Target Value
target == 0
target == 1
Density
CREDIT_INCOME_PERCENT

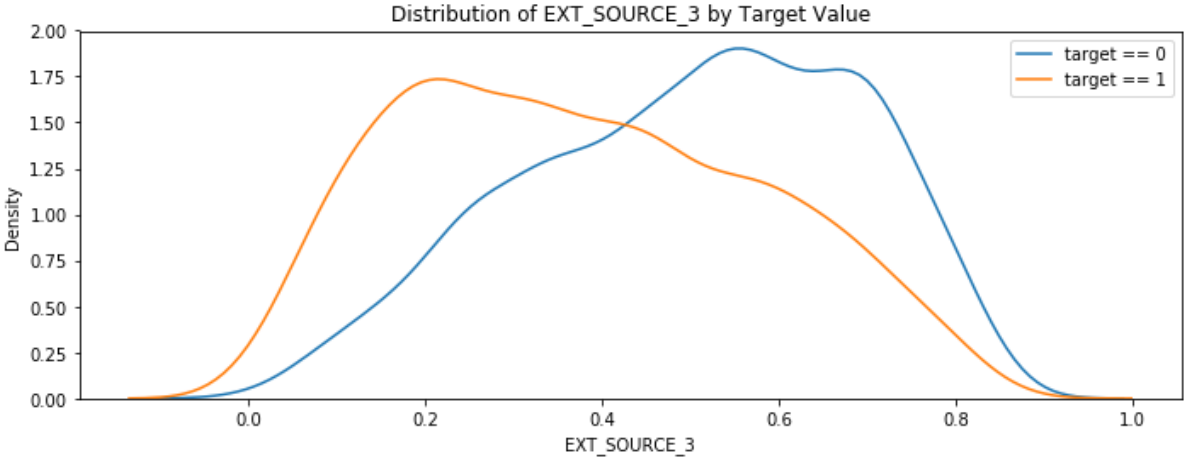

Distribution of ANNUITY_INCOME_PERCENT by Target Value
target == 0
target == 1
Density
ANNUITY_INCOME_PERCENT


Distribution of CREDIT_TERM by Target Value
target == 0
target == 1
Density
CREDIT_TERM

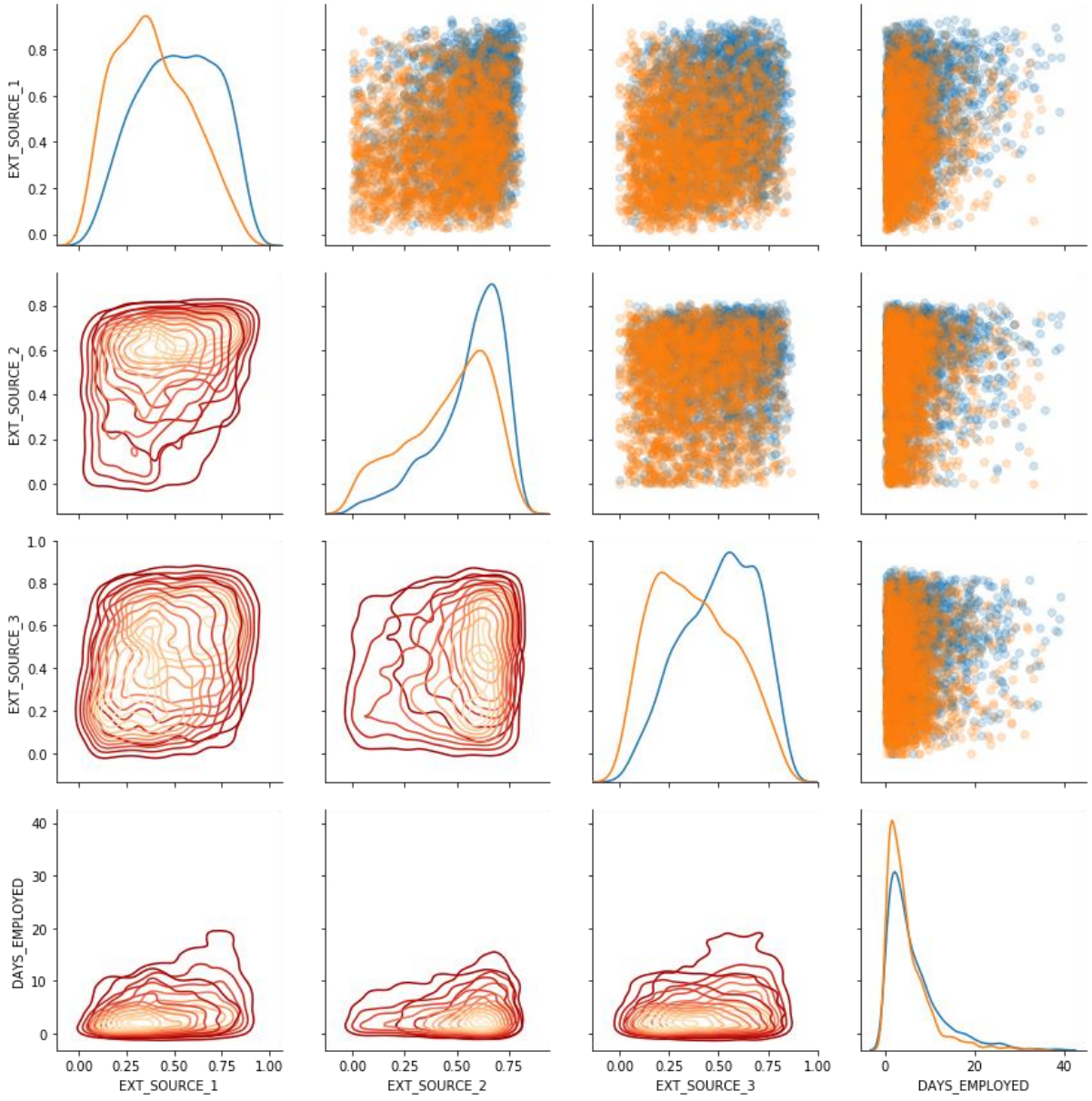

Distribution of DAYS_EMPLOYED_PERCENT by Target Value
target == 0
target == 1
Density
DAYS_EMPLOYED_PERCENT

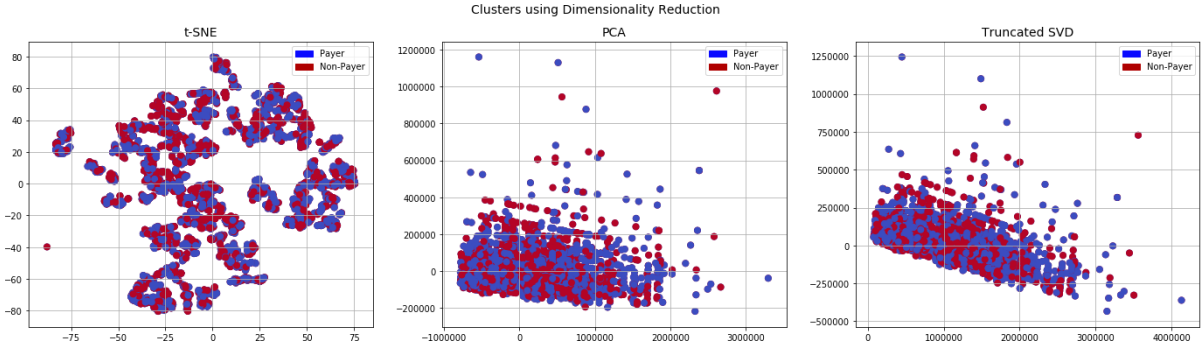

Distribution of EXT_SOURCE_1 by Target Value
target == 0
target == 1
Density
EXT_SOURCE_1

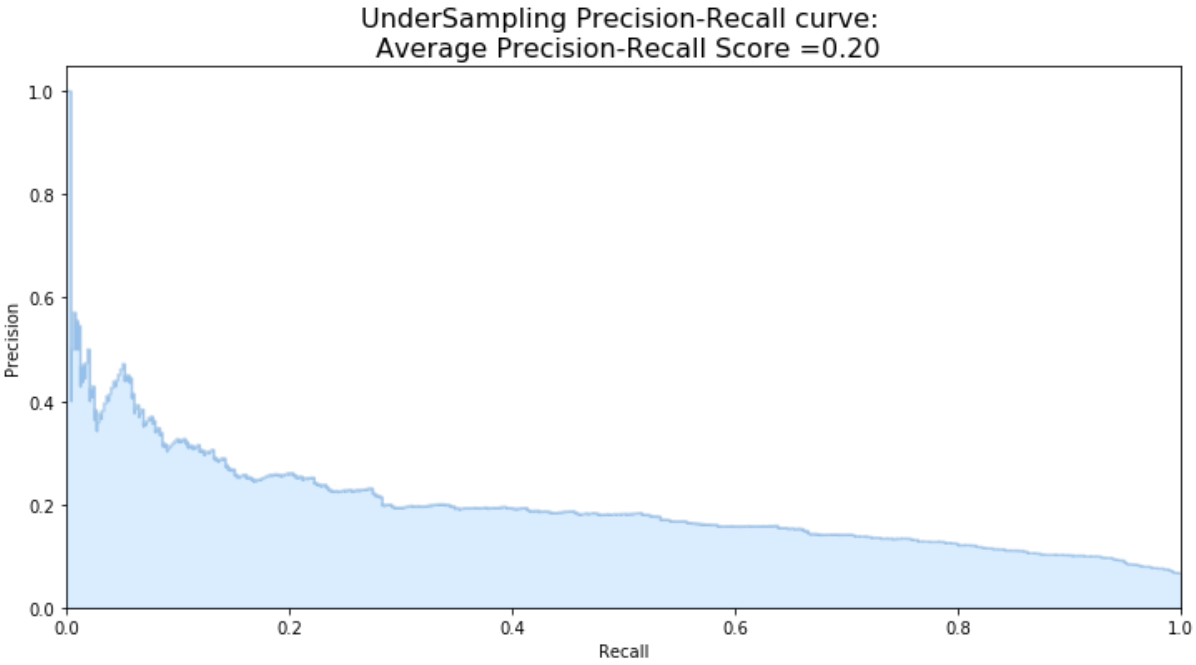

Distribution of EXT_SOURCE_2 by Target Value
target == 0
target == 1
Density
EXT_SOURCE_2

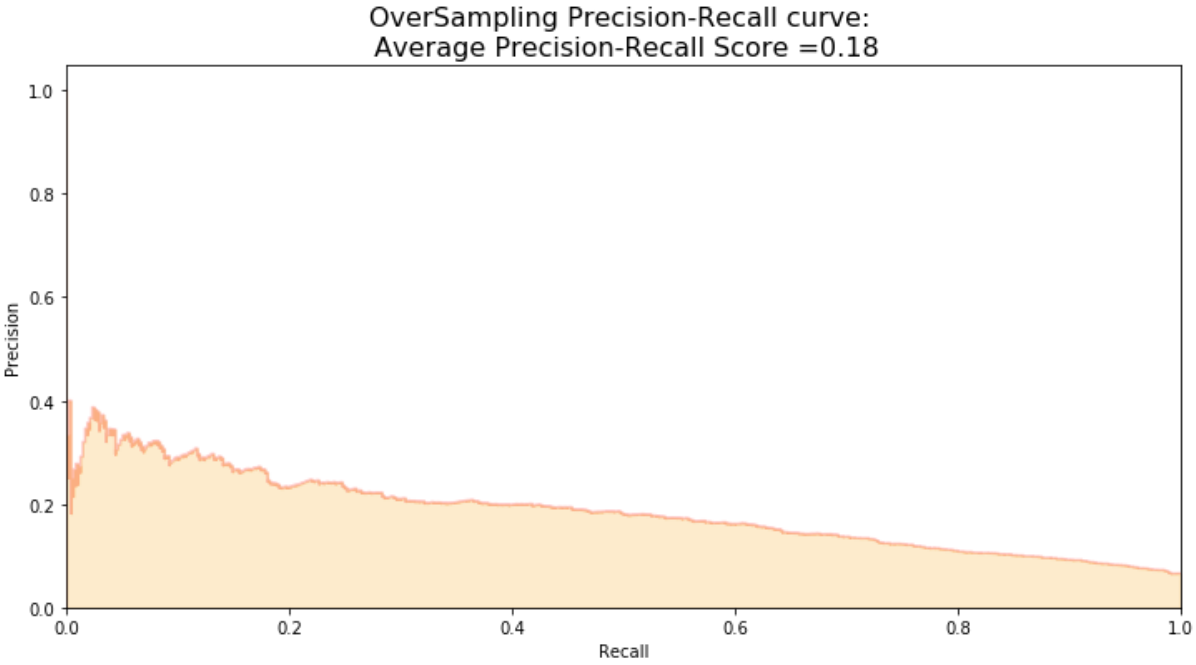

Distribution of EXT_SOURCE_3 by Target Value
target == 0
target == 1
Density
EXT_SOURCE_3

Ext Source and Days_Employed Features Pairs Plot






UnderSampling Precision-Recall curve:
Average Precision-Recall Score =0.20
Precision
Recall
0.0
0.2
0.4
0.6
0.8
1.0


OverSampling Precision-Recall curve:
Average Precision-Recall Score =0.18
Precision
Recall
0.0
0.2
0.4
0.6
0.8
1.0