1. Machine Learning Pipelines

https://github.com/yuxiaosun/capp-455136/tree/master/mlpipe. To import the pipeline, set the pulled directory into local path and use “from mlpipe import *”.

2. Data Input

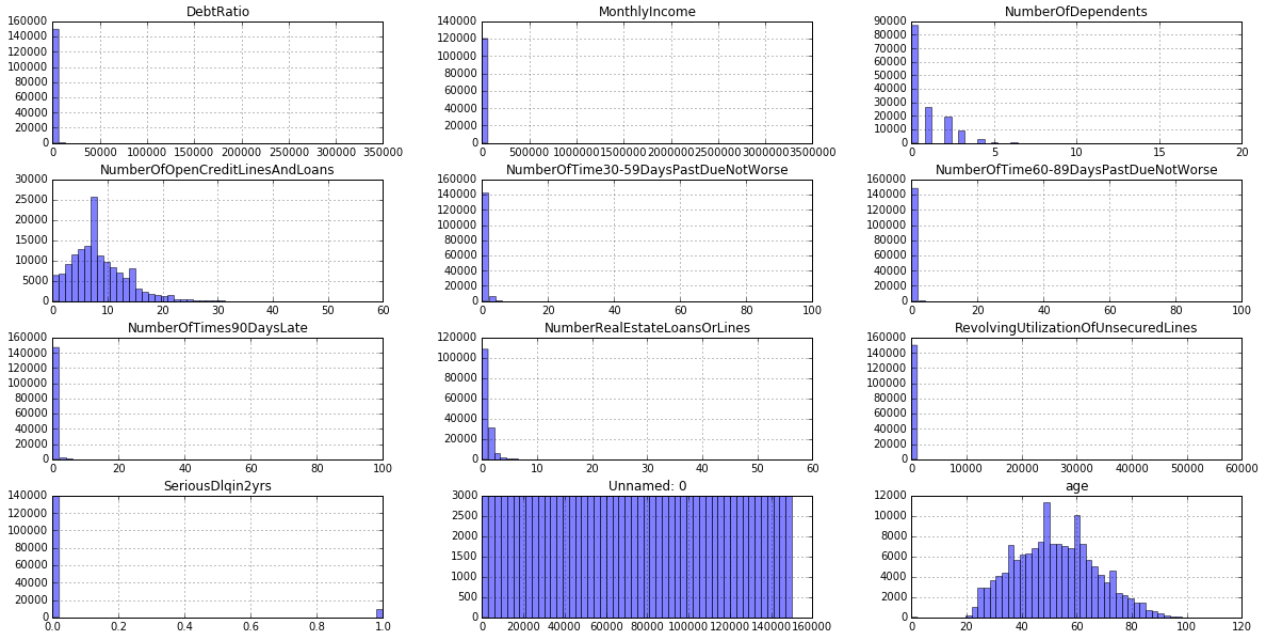
The data can be downloaded from https://www.kaggle.com/c/GiveMeSomeCredit/data

3. Data Exploration

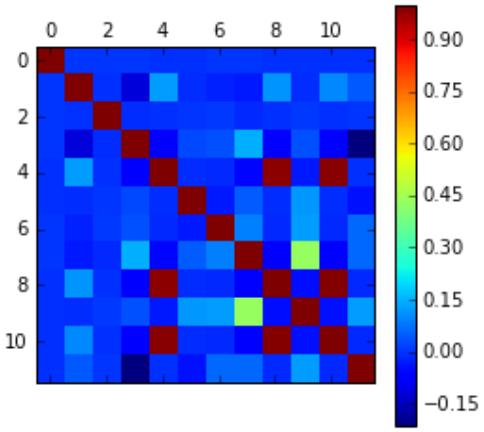
1) Summary Statistics

	mean	median	mode	std	missing
SeriousDlqin2yrs	0.07	0	0	0.25	0
RevolvingUtilizationOfUnsecuredLines	6.05	0.15	0	249.76	0
age	52.3	52	49	14.77	0
NumberOfTime30-59DaysPastDueNotWorse	0.42	0	0	4.19	0
DebtRatio	353.01	0.37	0	2037.82	0
MonthlyIncome	6670.22	5400	5000	14384.67	0.2
NumberOfOpenCreditLinesAndLoans	8.45	8	6	5.15	0
NumberOfTimes90DaysLate	0.27	0	0	4.17	0
NumberRealEstateLoansOrLines	1.02	1	0	1.13	0
NumberOfTime60-89DaysPastDueNotWorse	0.24	0	0	4.16	0
NumberOfDependents	0.76	0	0	1.12	0.03

2) Histograms



3) Cross-Correlation Map



Variable[0-10]=[Index, Serious Dlq in 2yrs, Revolving Utilization Of Unsecured Lines, age, Number Of Time 30-59 Days Past Due Not Worse, Debt Ratio, Monthly Income, Number Of Open Credit Lines And Loans, Number Of Times 90 Days Late, Number Real Estate Loans Or Lines, Number Of Time 60-89 Days Past Due Not Worse, Number Of Dependents]

Color indicates the corresponding Pearson correlation value.

4. Data Imputation

I impute the missing values of Monthly Income and Number of Dependents by KNN method, with K=1, features=two most correlated variables from the dataset.

5. Feature Engineering

Since the income distribution has long right tale, I discretize income variable into 15 income buckets and put monthly income greater than $15,000 into the same group. I also randomly split the dataset into training (80%) and testing (20%).

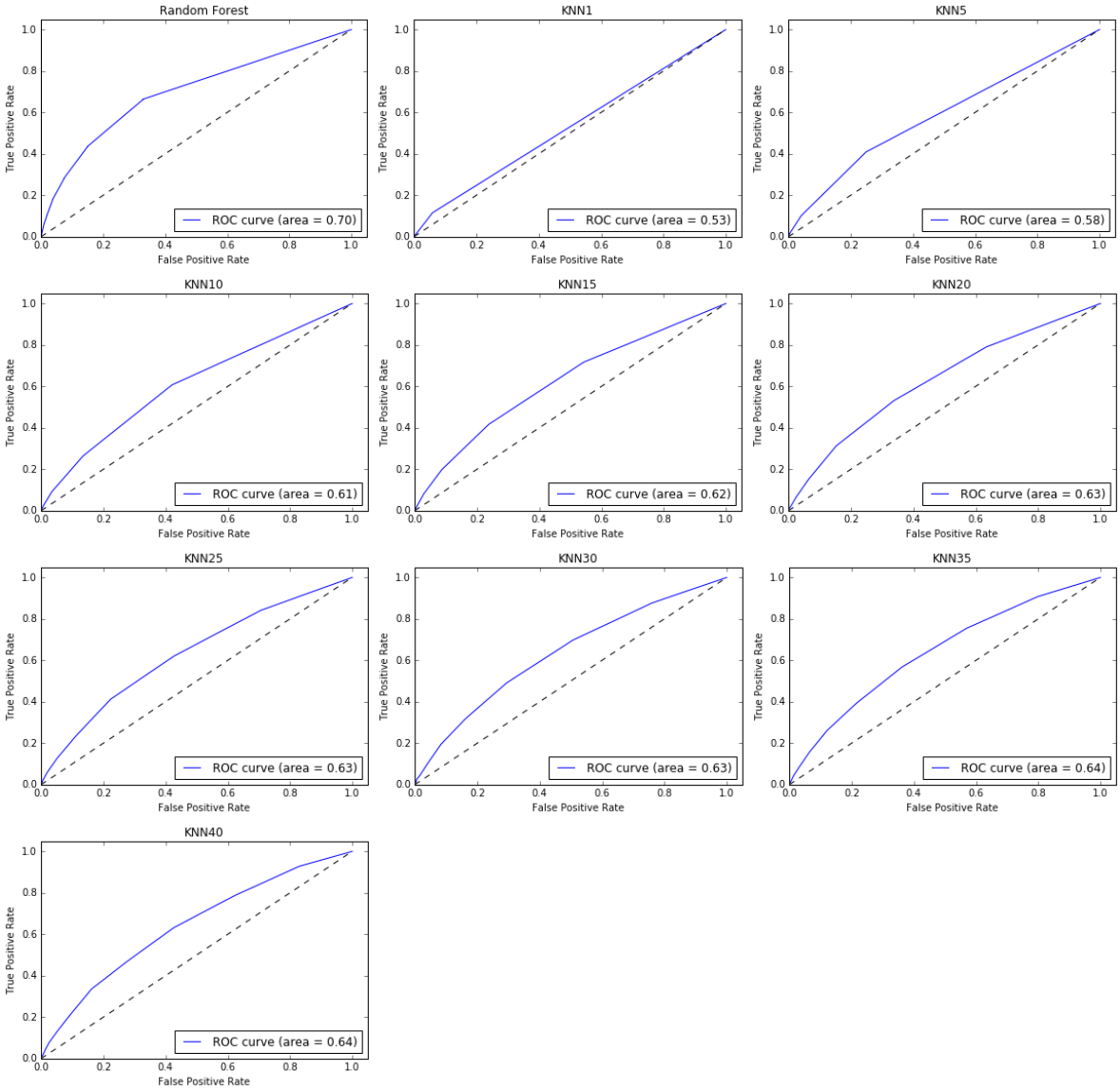
For feature selection, I use a Random Forest algorithm to determine which variables are the best at predicting risk: as the trees are randomly generated, the algorithm calculates how much worse a model does when each variable is left out.

6. Modelling

To predict risk, I use both KNN method (with k=1,5,10,15,20,25,30,35,40) and Random Forest Classifier.

7. Evaluation

I use ROC curve based on the testing data to evaluate different models.



Random Forest seems to be a better classifier in this case, which has the largest area and good ROC shape.

8. Application

Finally, I use the Random Forest classifier to predict missing risks values in the cs-test dataset. Click this link to see the result.