

# Midterm

STAT656

Spring 2018

Chris Berardi

Chien-Cheng Chen

David Sechrist

## Introduction

Our solution will be organized in the following manner. We will first discuss the data exploration portion of the SEMMA process, followed by the SAS solution, then the Python solution will be discussed. The approach to the problem for each language will be described, followed by the best model for each class. Best model means the optimum hyperparameter as selected by averaging over: accuracy, recall, precision, and F1 and selecting the highest average.

The best model from each class will then be used along with a 70/30 training validation split to determine the optimum solution for each language. Following the Python solution the two languages will be compared and an overall solution will be discussed.

In both methods default (default=1) is defined as the event in the models.

## Data Exploration

Before data are fit to any model, the data should first be explored to discover any sort of relationships between attributes, or between the attribute and the target. Also any missing or outlier values should also be identified for later imputation. While using the SAS Enterprise miner multiplot node, two graphs were of particular interest. Figure 1 shows June pay percent, that is, the percent of the credit from the last month that was paid off. As is clear from the graph, almost everyone either pays in full, or pays nothing or close to the minimum. It is also clear the incidence of default increases as the pay percent drops.

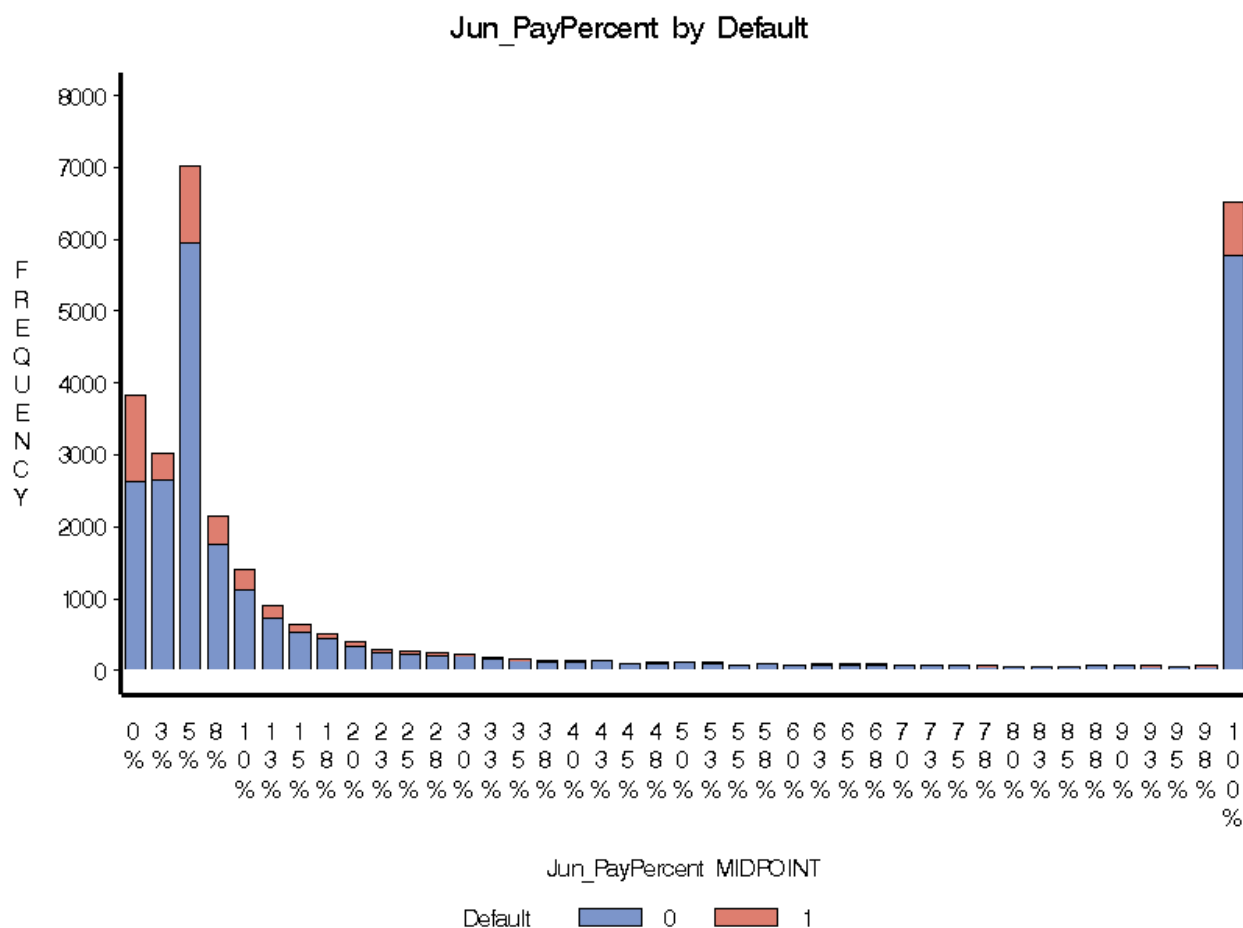


Figure 1

Figure 2 shows June status, which is the number of months a customer is behind in their payments. There is a clear relationship between being further behind on payment and default, which is not very surprising.

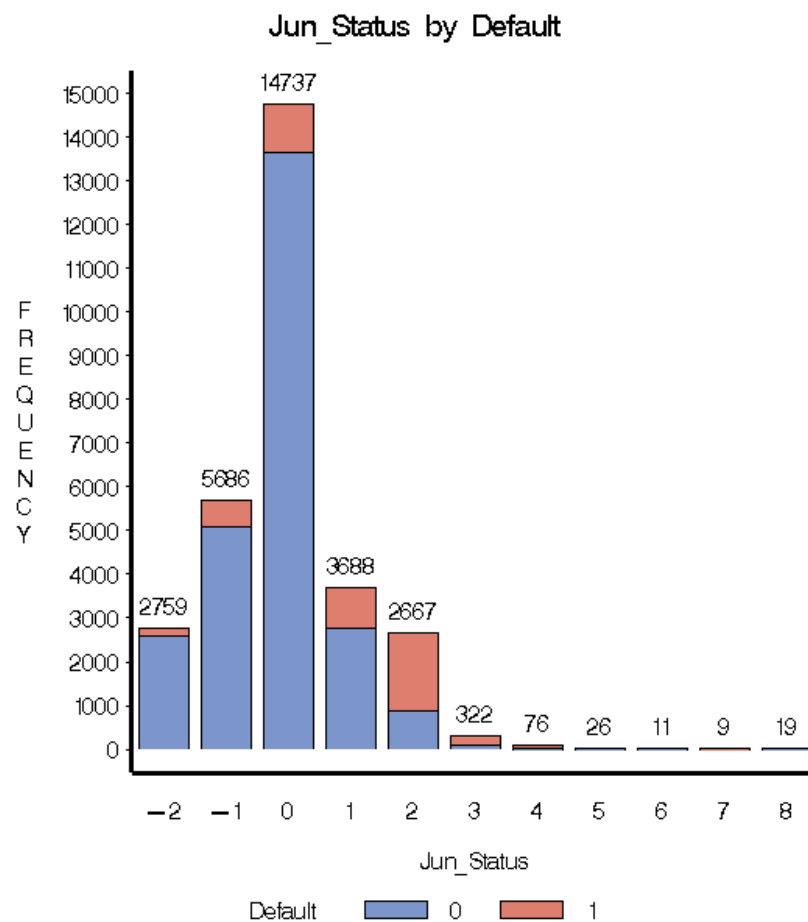


Figure 2

Both of these graphs together show that these attributes will most likely be significant in all of the models fit to the data.

Figure 3 shows the number of missing and outlier values for each attribute. Since none of the attributes is more than 25% missing, attributes will be imputed. Outliers will be set to missing and imputed as well.

Attribute Counts		
.....	Missing	Outliers
Default.....	0	0
Gender.....	3083	0
Education.....	4521	0
Marital_Status..	0	0
card_class.....	0	0
Age.....	5999	0
Credit_Limit....	0	0
Jun_Status.....	0	0
May_Status.....	0	0
Apr_Status.....	0	0
Mar_Status.....	0	0
Feb_Status.....	0	0
Jan_Status.....	0	0
Jun_Bill.....	0	1
May_Bill.....	0	1
Apr_Bill.....	0	1
Mar_Bill.....	0	0
Feb_Bill.....	0	0
Jan_Bill.....	0	1
Jun_Payment....	0	0
May_Payment....	0	0
Apr_Payment....	0	0
Mar_Payment....	0	0
Feb_Payment....	0	0
Jan_Payment....	0	0
Jun_PayPercent..	0	0
May_PayPercent..	0	0
Apr_PayPercent..	0	0
Mar_PayPercent..	0	0
Feb_PayPercent..	0	0
Jan_PayPercent..	0	0

Figure 3

## SAS

### Encoding

Following exploration of the data, missing values were imputed. Interval imputation was done with the tree method for both interval and nominal attributes.

### Random Forest

5 random forest models were fit to the data. One model used 50 trees with a sampling proportion of .6, the rest used 100 trees with sampling proportions of: .5, .6, .7, .8. The model using 100 trees and a proportion of .8 was found to obtain the best fit metrics. Variable importance was also determined and can be seen as Figure 4. As was seen in the data exploration, the status attributes are of great importance.

### Decision Tree

10 decision tree models were fit to the data. Non-HP models depths of: 5, 6, 7, 8, 9, 10, 12, 14, 16, were fit along with the following HP models: 5, 10, 15. 10-fold cross validation was done to select the model with the best fit statistics. The non-HP model with a depth of 16 was found to have the best statistics.

### Neural Network

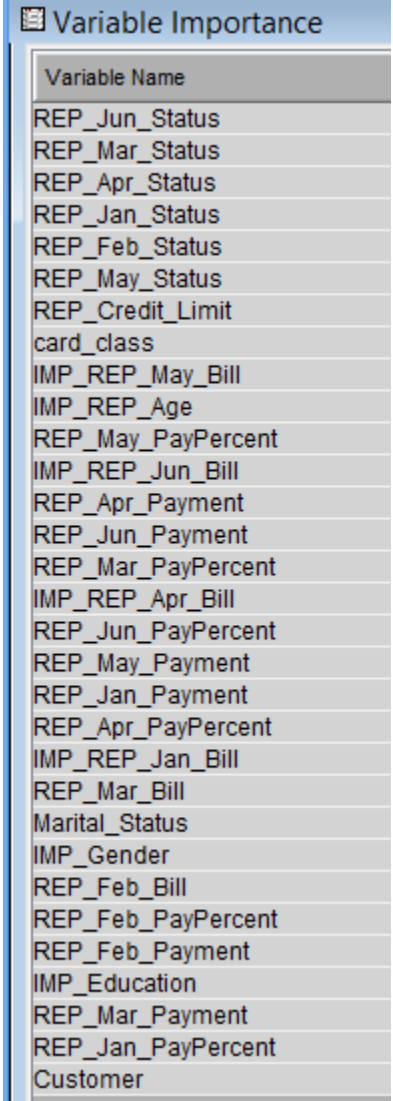
Before the neural network model was fit, all interval attributes were standardized. Following that 13 neural networks were fit to the data. Non-HP models fit were fit with the following number of perceptrons: 7, 9, 11, 13, 15. HP models were fit with the following configurations: 7, 9, 11, 13, (6,5), (7,6), (8,7), (9,8). 10-fold cross validation was used to assess each models fit and the model the optimum configuration was the (9,8) HP model.

### Logistic Regression

8 logistic regressions were run on the data. Both HP and non-HP models were fit using the full model, forward selection, backward selection, and stepwise selection. 10-fold cross validation was used to determine which gave the best results. The non-HP stepwise model resulted in the best fit.

The following 18 attributes were included in the best model:

Gender, Age, Marital\_Status, Apr\_Status, Credit\_Limit, Feb\_PayPercent, Feb\_Payment, Feb\_Status, Jan\_Status, Jun\_PayPercent, Jun\_Payment, Jun\_Status, Mar\_Bill, Mar\_PayPercent, May\_Pay\_Percent, May\_Payment, May\_Status, Card\_Class



Variable Name
REP_Jun_Status
REP_Mar_Status
REP_Apr_Status
REP_Jan_Status
REP_Feb_Status
REP_May_Status
REP_Credit_Limit
card_class
IMP_REP_May_Bill
IMP_REP_Age
REP_May_PayPercent
IMP_REP_Jun_Bill
REP_Apr_Payment
REP_Jun_Payment
REP_Mar_PayPercent
IMP_REP_Apr_Bill
REP_Jun_PayPercent
REP_May_Payment
REP_Jan_Payment
REP_Apr_PayPercent
IMP_REP_Jan_Bill
REP_Mar_Bill
Marital_Status
IMP_Gender
REP_Feb_Bill
REP_Feb_PayPercent
REP_Feb_Payment
IMP_Education
REP_Mar_Payment
REP_Jan_PayPercent
Customer

Figure 4

### Comparison of Model Types

Figure 5 shows the results of the 4 model classes fit to the data. The decision tree model produced the best fit, having the highest average for validation metrics. Furthermore none of the models are overfit, this can be seen because the difference between the training and validation metrics is not very large for any of the models.

	<b>Logistic</b>		<b>Neural Network</b>		<b>Decision Tree</b>		<b>Random Forest</b>	
<b>Hyperparameters</b>	Non-HP Stepwise		HP (9,8)		Non-HP Depth 16		100 Trees, Maximum Features .8	
	Training	Validation	Training	Validation	Training	Validation	Training	Validation
<b>Metric</b>								
<b>Misclassification</b>	0.140156	0.1347478	0.122232	0.126903	0.103677	0.112419	0.122661	0.126236
<b>Accuracy</b>	0.859844	0.8652522	0.877768	0.873097	0.896323	0.887581	0.877339	0.873764
<b>Recall</b>	0.274948	0.2916953	0.433951	0.404811	0.523109	0.505834	0.436305	0.418557
<b>Precision</b>	0.660537	0.7013201	0.696412	0.680925	0.761354	0.716229	0.692021	0.677419
<b>F1</b>	0.388277	0.4120213	0.534711	0.507759	0.620136	0.59292	0.535186	0.517417
<b>Mean(Excluding Misc.)</b>		0.5675722		0.616648		0.675641		0.621789

Figure 5

## Python

### Encoding

Following exploration of the data, missing values were imputed. Interval imputation was done with the mean value of the attribute, nominal imputation was done with the mode.

Three different imputation and encoding process were run. For all models one-hot encoding was used. For tree based models interval variables were not scaled and the final column for each nominal variable was not dropped. For neural network encoding interval variables were standardized and the final column was not dropped. For logistic regression interval variables were scaled and the final column was dropped.

All target Pandas data frames were converted to NumPy arrays to prevent warning messages from being triggered.

### Random Forest

The random forest results will be discussed first as they are used in the logistic regression solution. 12 different random forest models were tested for hyperparameter optimization. The following number of trees was used: 10, 20, 50. The percentage of features tested were: auto, .6, .7, .8.

The optimum model was found to have 50 trees with .8 of the features used. From this model a list of variable importance was extracted as Figure 6. As was discussed in the data exploration section, the status and payment percent attributes are of particular importance in the optimum random forest model.

### Decision Tree

6 decision tree models were fit to determine the optimum tree depth: 5, 6, 7, 8, 10, 12. 10-fold cross validation was used to determine the best model. The optimum depth was found to be 6.

### Neural Network

5 neural networks were fit to determine the optimum structure of the neural network. The following networks were fit: 3, 4, 5, (2,1), (2,2). 10-fold cross validation was used to determine the best model. Configuration (2,2) was found to have the best fit.

FEATURE.....	IMPORTANCE
Jun_Status.....	0.1975
May_Status.....	0.0634
Age.....	0.0527
Jun_Bill.....	0.0423
Credit_Limit....	0.0420
Jan_Bill.....	0.0353
Apr_Payment....	0.0346
Jan_Payment....	0.0329
May_Bill.....	0.0312
Apr_Bill.....	0.0307
Feb_Bill.....	0.0305
Mar_Payment.....	0.0299
May_Payment.....	0.0294
Jun_PayPercent..	0.0293
Jun_Payment.....	0.0293
Apr_PayPercent..	0.0288
Mar_Bill.....	0.0288
May_PayPercent..	0.0285
Mar_PayPercent..	0.0283
Jan_PayPercent..	0.0276
Feb_Payment.....	0.0269
Feb_PayPercent..	0.0265
Mar_Status.....	0.0148
Apr_Status.....	0.0140
Jan_Status.....	0.0121
Feb_Status.....	0.0105
Gender.....	0.0079
Education2.....	0.0056
Education3.....	0.0053
Marital_Status2.	0.0051
Marital_Status1.	0.0050
Education1.....	0.0050
card_class1.....	0.0027
Marital_Status3.	0.0017
card_class0.....	0.0016
card_class2.....	0.0013
Education5.....	0.0007
Marital_Status0.	0.0003
Education4.....	0.0002
Education6.....	0.0002
Education0.....	0.0000

Figure 6



## Logistic Regression

4 logistic models were fit to the data: full model, 20 predictors, 10 predictors, 5 predictors. Predictors were chosen from the ordering of attribute importance produces by the best random forest model. In order of importance the 20 most important predictors were found to be:

Jun\_Status, May\_Status, Age, Jun\_Bill, Credit\_Limit, Jan\_Payment, Jan\_Bill, Credit\_Limit, Jan\_Payment, Jan\_Bill, Mar\_Bill, Apr\_Payment, May\_Payment, May\_PayPercent, Apr\_Bill, Feb\_Bill, May\_Bill, Apr\_PayPercent, Feb\_PayPercent, Mar\_Payment, Jun\_Payment, Feb\_Payment, Mar\_PayPercent

10-fold cross validation was used to determine the best model. The model with the 20 most important predictors was found to have the best fit statistics.

## Comparison of Model Types

Comparing the best model from each model type, Figure 7 is obtained.

	<b>Logistic</b>		<b>Neural Network</b>		<b>Decision Tree</b>		<b>Random Forest</b>	
<b>Hyperparameters</b>	Predictors: 20		Configuration: (2,2)		Depth 6		50 Trees, Maximum Features .8	
	<b>Training</b>	<b>Validation</b>	<b>Training</b>	<b>Validation</b>	<b>Training</b>	<b>Validation</b>	<b>Training</b>	<b>Validation</b>
<b>Metric</b>								
<b>Misclassification</b>	0.136	0.136	0.123	0.128	0.119	0.126	0.001	0.12
<b>Accuracy</b>	0.8645	0.8643	0.8769	0.872	0.8806	0.8741	0.9995	0.8802
<b>Recall</b>	0.6938	0.6563	0.6846	0.6365	0.6877	0.6389	1	0.6758
<b>Precision</b>	0.3052	0.298	0.4573	0.4441	0.4929	0.4687	0.9968	0.4659
<b>F1</b>	0.4239	0.4099	0.5483	0.5232	0.5742	0.5407	0.9984	0.5516
<b>Mean(Excluding Misc.)</b>		0.557125		0.61895		0.6306		0.643375

Figure 7

While the random forest model has the highest metric average, the model is highly overfit. This can be seen in the large drop from the training to validation metrics. For that reason, even though it has slightly less optimum metrics, the decision tree is better—it does not suffer from the same overfitting issue.

## Comparison of SAS and Python Results

The SAS and the Python results, while quite different in models selected, are not very different in their fit metrics. The decision tree chosen by SAS was much deeper than the Python tree. The random forests were similar as well. The neural network chosen by SAS was much more complicated than the Python model.

In most cases SAS performed better than Python except for the neural network. This is similar to what was seen in the homework—a much simpler Python neural network outperformed a far more complicated SAS neural network. However, in all cases the metric averages were not very different between SAS and Python, within .05 of each other. So the average metric scores were similar for both languages. However there is one major difference: the SAS random forest is not severely overfitting the data, unlike the Python random forest.

In choosing the best model, the highest average metric score was obtained by the SAS non-HP Decision Tree with a depth of 16.