Once you have read the email, do research to figure out how you are going to do all of the work. Put a road map in place, install all packages you think you will need before you proceed. Let me know once you have read this big mail.

This is a problem from the credit card industry. These sort of data set are rarely available publicly and is a good opportunity to see how different machine learning techniques lends themselves to solving business problems. There are going to be several iterations, however getting the first cut of the codebase ready is important to be able to make tweaks and improve results. As we have discussed my motivation is for us to publish some of the results in a trade journal or a technical journal.

Now about the problem.
Banks decide who can get finance and on what terms and can make or break investment decisions. For markets and society to function, individuals and companies need access to credit.

Credit scoring algorithms, which make a guess at the probability of default, are the method banks use to determine whether or not a loan should be granted. The requirement is to predict the probability that somebody will experience financial distress in the next two years.

The goal is to build a model that borrowers can use to help make the best financial decisions.

DATA MANIPULATION:

RevolvingUtilizationOfUnsecuredLines (v1)
A. Notice that RevolvingUtilizationOfUnsecuredLines = 50708. But this should be a percentage (or rather, a fraction), so it should be at most 1. I assume that the actual value is 0.50708.
B. Apply above logic for all records value >= 2

age (v2)
A. with age>90 and 13 with age>100, the record is 109. Lower cap at 5 percentile. Upper cap at 95 pcntl

NumberOfTime30-59DaysPastDueNotWorse (v3)
A. You will see values abnormally high. 98,97 etc. These are some kind of codes. Create a new column called v3lab and set the values for the abnormally high values as 'abnormal' and for normal values as 'normal'

debt ratio (v4)
A. This should be typically less than 1. In many cases the data is missing or > 1. Create a new column v4lab, where for <1 label ' norm', > 1 label 'abnorm' for missing values (typically NA) say 'missing'
B. This is tied to the income field, which we will talk about later
C. It is likely that using v5 we have to impute this to get a more realistic debt ratio distribution

MonthlyIncome (v5)
A. Create a new variable v5d which will have the same values as in v5 but for missing values

will have 1
B. Create new variable monthly payments (n51)
C. The low income values were actually off by a factor of 1,000
D.

NumberOfOpenCreditLinesAndLoans (v6)
A. Nothing right now
NumberOfTimes90DaysLate (v7)
A. Nothing right now
NumberRealEstateLoansOrLines (v8)
A. Nothing
NumberOfTime60-89DaysPastDueNotWorse
A. Nothing
NumberOfDependents
A. Just backfill median value where missing

For each variable add a v*flg to identify the variable you have modified/added metadata in any way. I should like treatment done for both train and test data sets.

Next use cut function in R to create 10 buckets for each variable v*cut.

Once you have got this create a 10**10 grid of # samples, # occurrence and % occurrence in population. Drop all records with no values. Both for test and train. You can pull this into tableau to visualize this.

Do some box plot of the cuts to see how the occurrence vary. Graphics from the cookbook and the other cookbooks would be helpful

FEATURE SELECTION & CREATION
This is a important part of the work which require some investigation and creativity, in creating interaction variable that make business sense. This will be the next step. My experience- categorical variables are the best.

MODELING
1. The first thing to do is a simple logistics regression model
2. A simple decision tree model to build classification rules
3. A neural network model- 1/2 intermediate layer
3. Next is really uncharted territory:
    A. GBM, weighted GBMs,
    B. Random Forest, balanced Random Forest,
    C. GAM, weighted GAM (all with bernoulli/binomial error),
    D. SVM and bagged ensemble of SVMs

MODEL PERFORMANCE

Area Under the receiver operator Curve (AUC)

AUC is a commonly used evaluation method for binary choice problems, which involve classifying an instance as either positive or negative. Its main advantages over other evaluation methods, such as the simpler misclassification error, are:

It's insensitive to unbalanced datasets (datasets that have more installeds than not-installeds or vice versa).
For other evaluation methods, a user has to choose a cut-off point above which the target variable is part of the positive class (e.g. a logistic regression model returns any real number between 0 and 1 - the modeler might decide that predictions greater than 0.5 mean a positive class prediction while a prediction of less than 0.5 mean a negative class prediction). AUC evaluates entries at all cut-off points, giving better insight into how well the classifier is able to separate the two classes.
Understanding AUC

To understand the calculation of AUC, a few basic concepts must be introduced. For a binary choice prediction, there are four possible outcomes:

true positive - a positive instance that is correctly classified as positive;
false positive - a negative instance that is incorrectly classified as positive;
true negative - a negative instance that is correctly classified as negative;
false negative - a positive instance that is incorrectly classified as negative);
The true positive rate, or recall, is calculated as the number of true positives divided by the total number of positives. When identifying aircraft from radar signals, it is proportion that are correctly identified.

The false positive rate is calculated as the number of false positives divided by the total number of negatives. When identifying aircraft from radar signals, it is the rate of false alarms.

If somebody makes random guesses, the ROC curve will be a diagonal line stretching from (0,0) to (1,1) - see the blue line in the figure below. To understand this consider: Somebody who randomly guesses that 10 per cent of all radar signals point to planes. The false positive rate and the false alarm rate will be 10 per cent. Somebody who randomly guesses that 90 per cent of all radar signals point to planes. The false positive rate and the false alarm rate will be 90 per cent. Meanwhile a perfect model will achieve a true positive rate of 1 and a false positive rate of 0.

While ROC is a two-dimensional representation of a model's performance, the AUC distils this information into a single scalar. As the name implies, it is calculated as the area under the ROC curve. A perfect model will score an AUC of 1, while random guessing will score an AUC of around of 0.5. In practice, almost all models will fit somewhere in between

FOR YOUR REFERENCE
0 <==> 1 <==> 2 <==> 3 <==> 4 <==> 5 <==> 6 ==> Charge Off

A customer who keeps up with at least the monthly minimum payment will sit in the "0" bucket. However if she misses a payment, they would move to bucket "1". If she again misses another payment, she would be moved to bucket "2". And if she makes a payment to cover the 2 payments she has missed +the min. payment for the current month, she would be current and will move back to bucket "0".

If she misses more than 6 payments at a stretch , her account would be charged-off (note this does not mean the creditor will give up on getting their money back.) This is also called a serious delinquency. Every number relates to multiples of 30 as in the variable names mentioned.