Introduction:

Allstate is currently developing automated methods of predicting the cost, and hence severity, of claims. As a competitor, I attempt to create an algorithm which predicts the amount of the potential loss given a claim according to the features that they provide.Importantly, Allstate does not provide any information about what each feature represents, thus there will not be any description about the variables in the data set.
The data set is split randomly into two sets: train set and test set. The train set is used to train the model so that it can pick up the patterns of the data. Train and test set mostly have the same variables, however in the test set, the loss would be excluded and will not be given to the participant. Participants needs to use their model to predict the loss and submit to the competition. Kaggle will use mean absolute error, that is
to

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}\left|y_i - \hat{y}_i\right| = \frac{1}{n}\sum_{i=1}^{n}\left|e_i\right| .$$

Where

$$AE = |e_i| = |y_i - \hat{y}_i|$$

$$Actual = y_i$$

$$Predicted = \hat{y}_i$$

evaluate my prediction.

The Data:
In the train test, there are 131 columns of attributes, mostly categorical data (and the column labels conveniently tell you which data type they are), along with a response column called "loss". There are only 14 numerical variables in this data set. Each row is one insurance claim. My goal is to predict the loss for those claims.

Data Processing and Feature Engineering:

Since all the variables are anonymous, I will attempt to gain some clarity about the data for the purpose of feature engineering. I separate the data set into sub data sets, one is categorical and another one is continuous variables. The process will mainly try to solve the following questions

1. What can we do with anonymous categorical variables?
2. Is there any trend or relationship between continuous variables?

In order to answer the first questions, I attain few summary statistics such as the mode, levels, and the graph of each categorical variables. Luckily most of categorical variables in this data set

only have two level values. Thus, when transforming the categorical variables into numerical values, I won't add too many dimensions to the data set.

Out[8]:

	cat1	cat2	cat3	cat4	cat5	cat6	cat7	cat8
count	188318	188318	188318	188318	188318	188318	188318	188318
unique	2	2	2	2	2	2	2	2
top	A	A	A	A	A	A	A	A
freq	141550	106721	177993	128395	123737	131693	183744	177274

	cat9	cat10	...	cat107	cat108	cat109	cat110	cat111
count	188318	188318	...	188318	188318	188318	188318	188318
unique	2	2	...	20	11	84	131	16
top	A	A	...	F	B	BI	CL	A
freq	113122	160213	...	47310	65512	152918	25305	128395

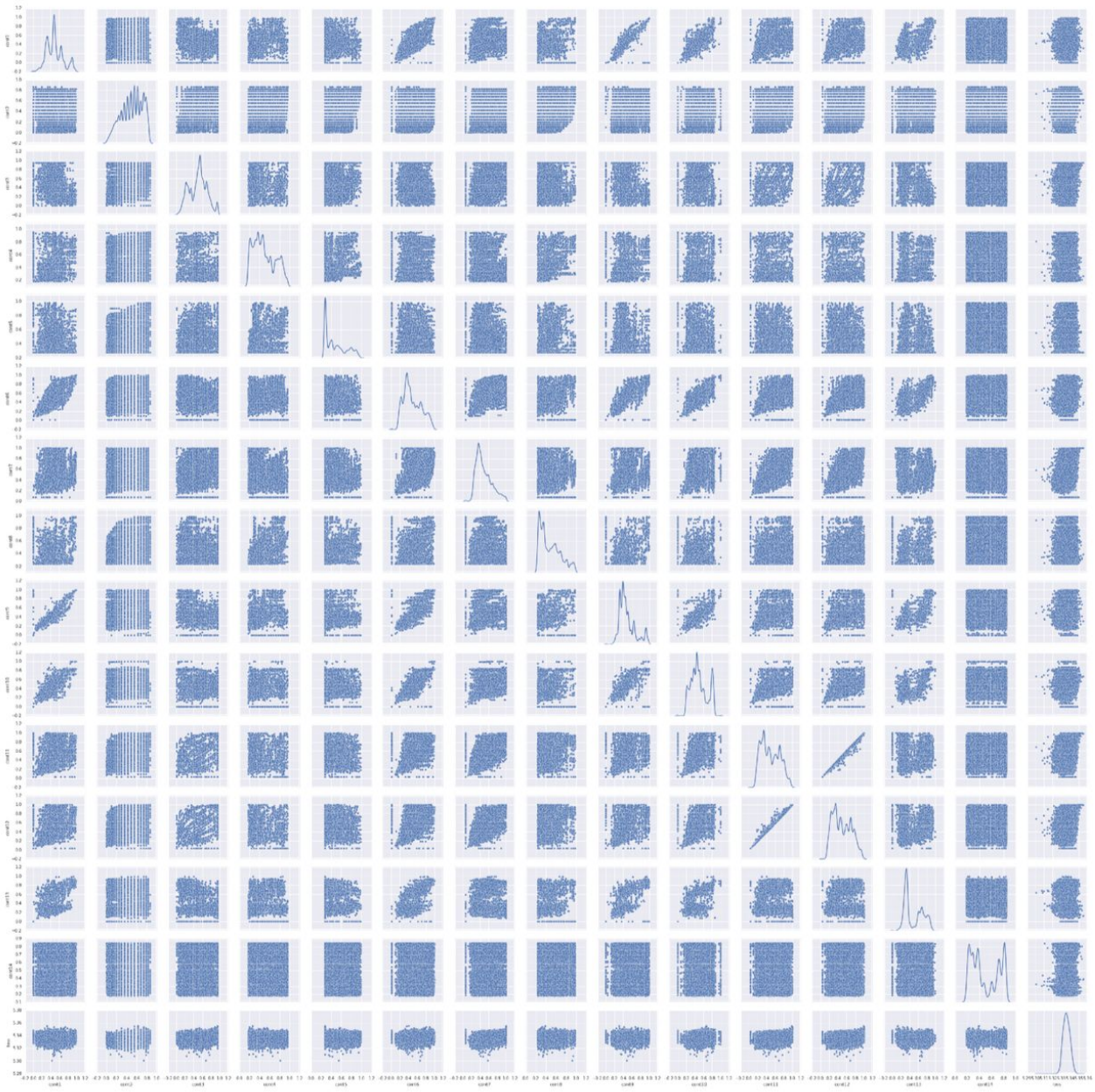
	cat112	cat113	cat114	cat115	cat116
count	188318	188318	188318	188318	188318
unique	51	61	19	23	326
top	E	BM	A	K	HK
freq	25148	26191	131693	43866	21061

However, even with the help with visualization and descriptive statistics, it is impossible to pick up any trend or meaning of those variables. Thus I will simply rely on python's random forest feature importance in order to pick up the most importants variables. Another note is that one variables has 50 levels, where the values seems to be an abbreviation of US state. This can be one useful point for feature engineering, such as matching the state with the code and add one variables that keeps the population of that state. It is natural to think that the state with high population will have higher loss, thus it can improve the accuracy of the model.

I have more rooms to answer question 2. Beside from the distribution plot of each variable, I decided to plot two variables against each other to check if there is any regression relationship, since it is the easiest and simplest check.

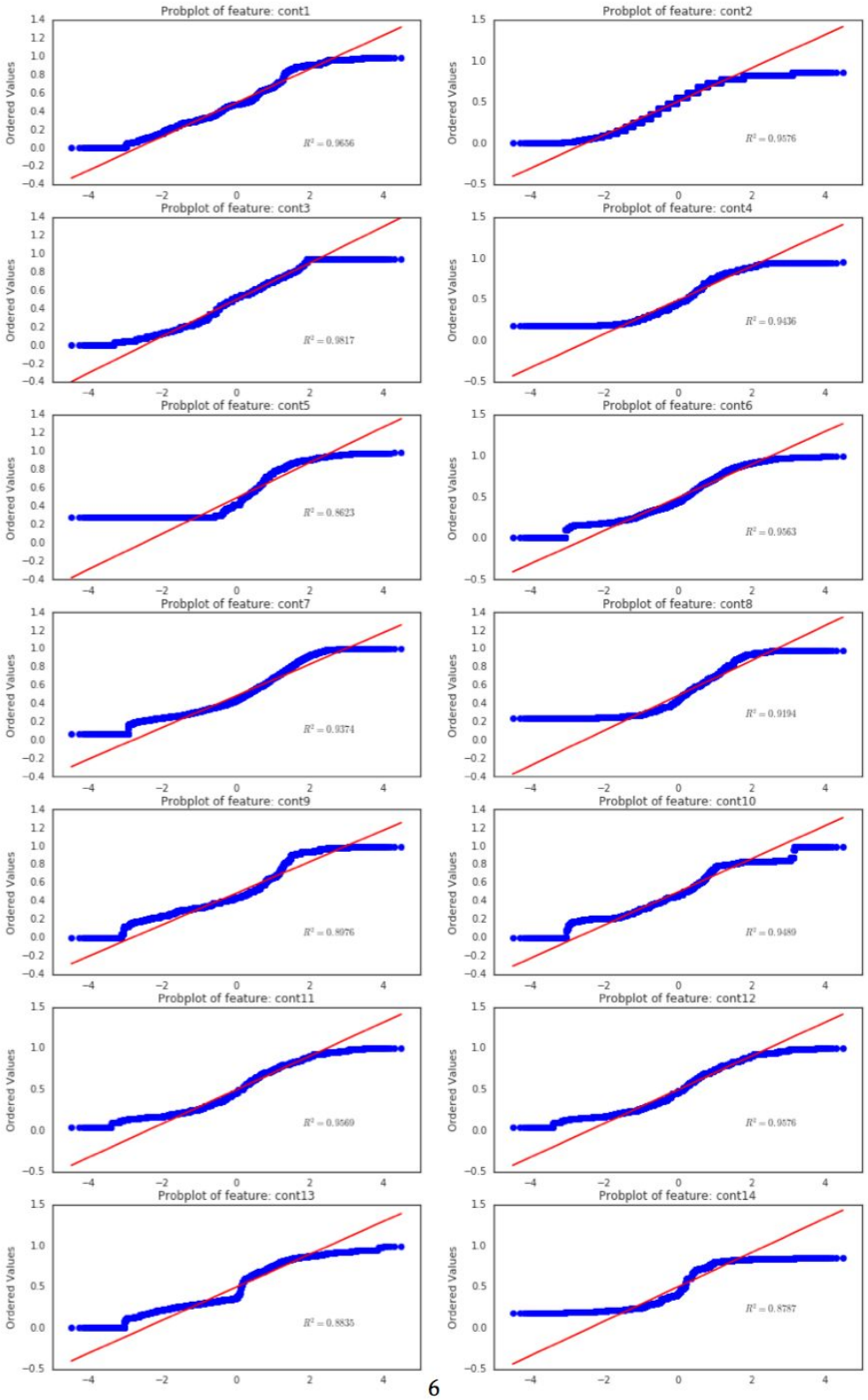
Here are two observations from the plots:

1. Most of continuous variables don't have a normal distribution.
2. There are obvious regression relationship between some pairs of variables, such as cont 9 vs cont 10; cont 6 vs cont 7 etc...



The first observation gives me a direction to the next step: transforming the original distribution of each variable to normal distribution. This step is very crucial since most statistical models depend on assumption of normality. To check if the my choice of transformation is reasonable, I used qq-plot to compare the distribution. Before start this experiment, I quickly check the skewness of each probability plot. The threshold is 0.25, thus any continuous variable with skew greater than the threshold will be included in the experiment. Using python, I got the list of 12 out of 14 variables that need changes. In the end, I

choose box-cox, log, tan, power transformation for my continuous variables data set. Rechecking qq-plot against Gaussian distribution, the data point seems to go along the a straight linear line. I attain the conclusion that my transformation is reasonable.



Since there is a regression relationship between some variables, I come up with another strategy to improve my prediction models: using ensembling technique. Ensemble will allow me to combine different statistical models in one data set to get the best result. To do ensemble, I would need to slip the data set into k fold, use one model on each fold, predict the result of each fold, and average k-predictions to get the final prediction. Since the dataset is anonymized, it is hard to use only one model to pick up the trend in the data sets. By using multiple models, I would be able to get the best result. The second observation suggests that linear regression can be one of the models. The other models would random forest, k-near neighbor, and XGB Boost. Another thing to keep in mind this data set contains almost 200000 records, thus using too many models would exhaust the computation power.

Outcome

To compare the significance of these strategies, I turned in multiple submissions that implement or don't implement any change. The transformation of continuous variables alone move me up by 10 ranks in the leader board. For the second strategy, after trying different k value, I decided to split dataset into 4 folds only. This improves my score by 2%. Combining both strategies, my score has improved by 5%.