I refined the model one more time, and this file will explain the model or some other code in detail.

* First, the visualization of missing part.

图形用户界面, 文本

描述已自动生成

Parameter explanation: figsize = (width, length)

cmap = Color Brewer palettes

图示

描述已自动生成

Plot explanation: Left numbers represent row number, bottom represents the variables. Dark blue represents the missing part. So we can obviously see the NA distribution of the dataset.

* The reason why we need convert the type is, only categorical or numeric variables can compute PCC between each other.

文本

描述已自动生成

* About correlation table

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

We create a table to record the PCC between each incomplete var and complete var.

表格

描述已自动生成

Have 13 complete and 35 incomplete vars.

For complete vars, every 13 rows will be the same.

For incomplete vars, 0-12 rows is the first incomplete var, 13-25 rows is the second incomplete var ``````

Can use retrieve method to split the table into 35 sub-table, and sort the values to find out the most correlated complete vars for each incomplete var.

文本

描述已自动生成

* In the test\_model function, ‘n\_splits=10’ determines 10 different validation (and training) sets will create, and it returns index of 10 trainset and test set.
* df\_filter is the subset of df which contains all rows with no missing in ‘i’, so for each ‘i’ the df\_filter is different, trainset and test set are both subset of df\_filter.
* df\_test is test set, df\_train is trainset. X\_train is dependent variables in df\_train (in this model, features columns are X\_train), while y\_train is independent variable in df\_train. We use X\_train and y\_train to fit model (use features to predict ‘i’). X\_test is input value, and add X\_test into model.predict() can get predicted output value (which is expected more access to y\_test more better), y\_test is the true value. So finally we compare predict value with true value, and use statistical methods to evaluate the model.
* I changed the indicator to evaluate model.

I used MSE to evaluate before, but actually MSE is more useful when compare several models. Now I use R squared score instead.

R-squared (R^2) is a statistical measure that represents the proportion of the variance for a dependent variable that's explained by an independent variable or variables in a regression model.—Wiki

- if  R-squared value < 0.3 this value is generally considered a None or Very weak effect size,

- if R-squared value 0.3 < r < 0.5 this value is generally considered a weak or low effect size,

- if R-squared value 0.5 < r < 0.7 this value is generally considered a Moderate effect size,

- if R-squared value r > 0.7 this value is generally considered strong effect size.

So I select incomplete variables with r2 score > 0.6 to do imputation by model, and for variables with r2 score < 0.6, I dropped 3 variables with high ratio of missing values, and imputed NAs by Area mean for other variables which have less NAs.

图形用户界面, 文本, 应用程序

描述已自动生成

* For impute\_NA function, I insert the model in it. For X\_train and y\_train, they are still the same as test\_model() function. But the test set is changed to df\_missing (the subset of df that ‘i’ is entirely missing in the set, but features (X\_test) are complete). So we use predicted value (model.predict(X\_test.)) to impute true value (y\_test).