

Untitled

February 8, 2017

```
In [7]: import datetime
import numpy as np
import pandas as pd

In [8]: from sklearn.metrics import mean_squared_error
from sklearn.model_selection import KFold
import xgboost as xgb
from sklearn.linear_model import Lasso, Ridge, ElasticNet
from sklearn.kernel_ridge import KernelRidge
from sklearn.base import BaseEstimator, RegressorMixin

In [9]: from sklearn.preprocessing import LabelEncoder
from scipy.stats import skew

In [10]: start_time = datetime.datetime.now()

In [11]: class CustomEnsembleRegressor(BaseEstimator, RegressorMixin):
    def __init__(self, regressors = None):
        self.regressors = regressors

    def fit(self, X, y):
        for regressor in self.regressors :
            regressor.fit(X,y)

    def predict(self , X):
        self.predictions_ = list()
        for regressor in self.regressors :
            self.predictions_.append(np.exp(regressor.predict(X).ravel()))

        return np.log1p(np.mean(self.predictions_ , axis=0))

    return mproba

In [12]: def rmse(y_true, y_pred):
    return np.sqrt(mean_squared_error(y_true, y_pred))

In [13]: def factorize(df, factor_df, column, fill_na= None):
    factor_df[column]=df[column]
```

```
if fill_na is not None:
    factor_df[column].fillna(fill_na, inplace=1)
    le.fit(factor_df[column].unique())
    factor_df[column] = le.transform(factor_df[column])
return factor_df
```

Now I'm going to combine all numerical features into one big DataFrame.

In []: