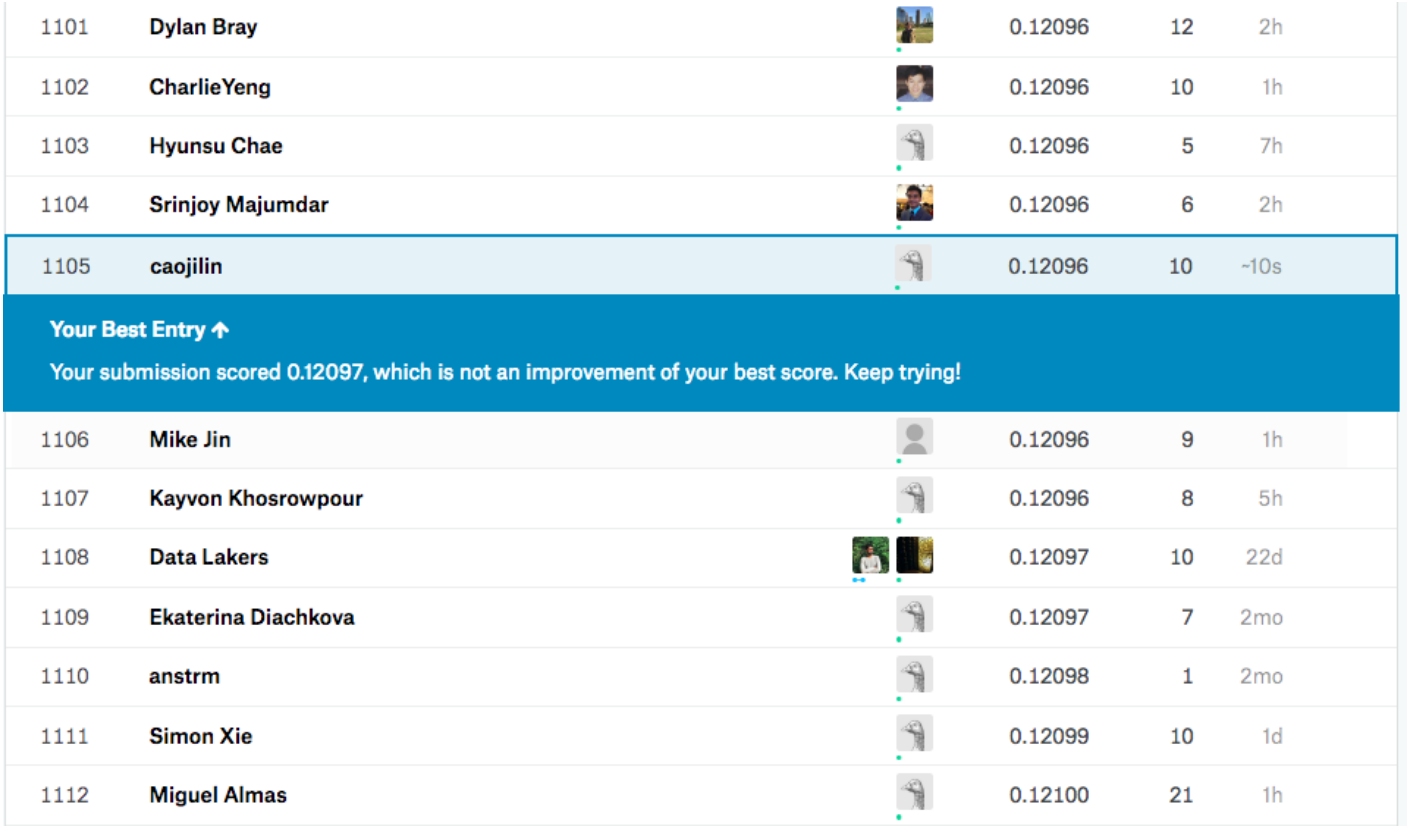
Homework 4 - Berkeley STAT 157

Your name: cao jilin, SID 3033278367, teammates Mike Jin, Daniel Kim (Please add your name, SID and teammates to ease Ryan and Rachel to grade.)



1101	Dylan Bray		0.12096	12	2h
1102	CharlieYeng		0.12096	10	1h
1103	Hyunsu Chae		0.12096	5	7h
1104	Srinjoy Majumdar		0.12096	6	2h
1105	caojilin		0.12096	10	~10s

Your Best Entry ↑

Your submission scored 0.12097, which is not an improvement of your best score. Keep trying!

1106	Mike Jin		0.12096	9	1h
1107	Kayvon Khosrowpour		0.12096	8	5h
1108	Data Lakers		0.12097	10	22d
1109	Ekaterina Diachkova		0.12097	7	2mo
1110	anstrm		0.12098	1	2mo
1111	Simon Xie		0.12099	10	1d
1112	Miguel Almas		0.12100	21	1h

In [76]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib

import d2l

import matplotlib.pyplot as plt
from scipy.stats import skew
from scipy.stats.stats import pearsonr
from mxnet.gluon import nn
from mxnet import autograd, nd
from mxnet.gluon import data as gdata
from mxnet import init
from mxnet.gluon import loss as gloss
from mxnet import gluon
```

In [3]:
```
train = pd.read_csv("kaggle_house_pred_train.csv")
test = pd.read_csv("kaggle_house_pred_test.csv")
```

In [4]:
```
all_data = pd.concat((train.loc[:,'MSSubClass':'SaleCondition'],
                      test.loc[:,'MSSubClass':'SaleCondition']))
```

In [5]:
```
train["SalePrice"] = np.log1p(train["SalePrice"])
```

In [6]:
```
numeric_feats = all_data.dtypes[all_data.dtypes != "object"].index

skewed_feats = train[numeric_feats].apply(lambda x: skew(x.dropna())) #c
ompute skewness
skewed_feats = skewed_feats[skewed_feats > 0.75]
skewed_feats = skewed_feats.index

all_data[skewed_feats] = np.log1p(all_data[skewed_feats])
skewed_feats
```

Out[6]:
```
Index(['MSSubClass', 'LotFrontage', 'LotArea', 'MasVnrArea', 'BsmtFinSF
1',
       'BsmtFinSF2', 'BsmtUnfSF', 'TotalBsmtSF', '1stFlrSF', '2ndFlrS
F',
       'LowQualFinSF', 'GrLivArea', 'BsmtHalfBath', 'KitchenAbvGr',
       'WoodDeckSF', 'OpenPorchSF', 'EnclosedPorch', '3SsnPorch',
       'ScreenPorch', 'PoolArea', 'MiscVal'],
      dtype='object')
```

In [7]:
```
all_data = pd.get_dummies(all_data)
```

In [8]:
```
all_data = all_data.fillna(all_data.mean())
```

In [9]:
```
X_train = all_data[:train.shape[0]]
X_test = all_data[train.shape[0]:]
y = train.SalePrice
```

In [10]:
```
from sklearn.linear_model import ElasticNet, LassoCV, LassoLarsCV
from sklearn.model_selection import cross_val_score
```

In [11]:
```
def rmse_cv(model):
    rmse= np.sqrt(-cross_val_score(model, X_train, y, scoring="neg_mean_
squared_error", cv = 5))
    return(rmse)
```

In [89]:
```
model_lasso = LassoCV(alphas = [1, 0.1, 0.001, 0.0005], cv=5).fit(X_trai
n, y)
```

In [90]:
```
rmse_cv(model_lasso).mean()
```

Out[90]:
```
0.12256735885048149
```

In [14]:
```
coef = pd.Series(model_lasso.coef_, index = X_train.columns)
```

In [15]:
```
coef[coef !=0 ].index
```

Out[15]:
```
Index(['MSSubClass', 'LotArea', 'OverallQual', 'OverallCond', 'YearBuil
t',
       'YearRemodAdd', 'MasVnrArea', 'BsmtFinSF1', 'BsmtFinSF2', 'BsmtU
nfSF',
       ...
       'GarageQual_Fa', 'GarageCond_Fa', 'PavedDrive_Y', 'Fence_GdWo',
       'SaleType_COD', 'SaleType_New', 'SaleType_WD', 'SaleCondition_Ab
norml',
       'SaleCondition_Family', 'SaleCondition_Normal'],
      dtype='object', length=111)
```

In [16]:
```
lasso_preds = np.expm1(model_lasso.predict(X_test))
```

In [17]:
```
solution = pd.DataFrame({"id":test.Id, "SalePrice":lasso_preds})
solution.to_csv("submission.csv", index = False)
```