benz project

January 15, 2022

[2]:
```
import pandas as pd
```

[3]:
```
df_test=pd.read_csv('test.csv')
```

[4]:
```
df_train=pd.read_csv('train.csv')
```

[5]:
```
df_test
```

[5]:

	ID	X0	X1	X2	X3	X4	X5	X6	X8	X10	…	X375	X376	X377	X378	\
0	1	az	v	n	f	d	t	a	w	0	…	0	0	0	1	
1	2	t	b	ai	a	d	b	g	y	0	…	0	0	1	0	
2	3	az	v	as	f	d	a	j	j	0	…	0	0	0	1	
3	4	az	l	n	f	d	z	l	n	0	…	0	0	0	1	
4	5	w	s	as	c	d	y	i	m	0	…	1	0	0	0	
…	…	..	..	..	..	..	..	..	..	…	…	…	…	…	…	
4204	8410	aj	h	as	f	d	aa	j	e	0	…	0	0	0	0	
4205	8411	t	aa	ai	d	d	aa	j	y	0	…	0	1	0	0	
4206	8413	y	v	as	f	d	aa	d	w	0	…	0	0	0	0	
4207	8414	ak	v	as	a	d	aa	c	q	0	…	0	0	1	0	
4208	8416	t	aa	ai	c	d	aa	g	r	0	…	1	0	0	0	

	X379	X380	X382	X383	X384	X385
0	0	0	0	0	0	0
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0
…	…	…	…	…	…	…
4204	0	0	0	0	0	0
4205	0	0	0	0	0	0
4206	0	0	0	0	0	0
4207	0	0	0	0	0	0
4208	0	0	0	0	0	0

[4209 rows x 377 columns]

[6]:
```
df_train
```

[6]:
```
      ID       y  X0 X1  X2 X3 X4  X5 X6 X8  …  X375  X376  X377  X378  \
0      0  130.81   k  v  at  a  d   u  j  o  …     0     0     1     0
1      6   88.53   k  t  av  e  d   y  l  o  …     1     0     0     0
2      7   76.26  az  w   n  c  d   x  j  x  …     0     0     0     0
3      9   80.62  az  t   n  f  d   x  l  e  …     0     0     0     0
4     13   78.02  az  v   n  f  d   h  d  n  …     0     0     0     0
…    …      …  .. ..  .. .. ..  .. .. ..  …   …   …   …   …
4204  8405  107.39  ak  s  as  c  d  aa  d  q  …     1     0     0     0
4205  8406  108.77   j  o   t  d  d  aa  h  h  …     0     1     0     0
4206  8412  109.22  ak  v   r  a  d  aa  g  e  …     0     0     1     0
4207  8415   87.48  al  r   e  f  d  aa  l  u  …     0     0     0     0
4208  8417  110.85   z  r  ae  c  d  aa  g  w  …     1     0     0     0

      X379  X380  X382  X383  X384  X385
0        0     0     0     0     0     0
1        0     0     0     0     0     0
2        0     0     1     0     0     0
3        0     0     0     0     0     0
4        0     0     0     0     0     0
…    …   …   …   …   …   …
4204     0     0     0     0     0     0
4205     0     0     0     0     0     0
4206     0     0     0     0     0     0
4207     0     0     0     0     0     0
4208     0     0     0     0     0     0

[4209 rows x 378 columns]
```

[7]:
```
y_train=df_train['y'].values
```

[8]:
```
y_train
```

[8]:
```
array([130.81,  88.53,  76.26, …, 109.22,  87.48, 110.85])
```

[9]:
```
cols = [c for c in df_train.columns if 'X' in c]
print('Number of features: {}'.format(len(cols)))
```

```
Number of features: 376
```

[10]:
```
df_train[cols].dtypes.value_counts()
```

[10]:
```
int64     368
object      8
dtype: int64
```

[11]:
```
import numpy as np
```

[12]:
```
counts = [[], [], []]
for c in cols:
    typ = df_train[c].dtype
    uniq = len(np.unique(df_train[c]))
    if uniq == 1:
        counts[0].append(c)
    elif uniq == 2 and typ == np.int64:
        counts[1].append(c)
    else:
        counts[2].append(c)
```

[13]:
```
print('Constant features: {} Binary features: {} Categorical features: {}'.
 ↪format(*[len(c) for c in counts]))
print('Constant features:', counts[0])
print('Categorical features:', counts[2])
```

```
Constant features: 12 Binary features: 356 Categorical features: 8
Constant features: ['X11', 'X93', 'X107', 'X233', 'X235', 'X268', 'X289',
'X290', 'X293', 'X297', 'X330', 'X347']
Categorical features: ['X0', 'X1', 'X2', 'X3', 'X4', 'X5', 'X6', 'X8']
```

[14]:
```
usable_columns = list(set(df_train.columns) - set(['ID', 'y']))
y_train = df_train['y'].values
id_test = df_test['ID'].values

x_train = df_train[usable_columns]
x_test = df_test[usable_columns]
```

[15]:
```
def check_missing_values(df):
    if df.isnull().any().any():
        print("There are missing values in the dataframe")
    else:
        print("There are no missing values in the dataframe")
check_missing_values(x_train)
check_missing_values(x_test)
```

```
There are no missing values in the dataframe
There are no missing values in the dataframe
```

[16]:
```
for column in usable_columns:
    cardinality = len(np.unique(x_train[column]))
    if cardinality == 1:
        x_train.drop(column, axis=1)
        x_test.drop(column, axis=1)
    if cardinality > 2:
        mapper = lambda x: sum([ord(digit) for digit in x])
        x_train[column] = x_train[column].apply(mapper)
```

```
        x_test[column] = x_test[column].apply(mapper)
x_train.head()
```

```
<ipython-input-16-331f1819b51b>:8: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  x_train[column] = x_train[column].apply(mapper)
<ipython-input-16-331f1819b51b>:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  x_test[column] = x_test[column].apply(mapper)
```

[16]:

	X91	X283	X66	X341	X308	X187	X108	X346	X51	X365	…	X44	X273	\
0	0	0	0	0	0	1	0	0	0	0	…	0	1	
1	0	0	0	0	0	1	0	0	1	0	…	0	1	
2	0	0	0	0	0	0	0	0	1	0	…	0	1	
3	0	0	0	0	0	0	1	0	0	0	…	0	1	
4	0	0	0	0	0	0	1	0	1	0	…	0	1	

	X16	X350	X113	X179	X277	X318	X248	X48
0	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	0	0
2	0	1	0	1	0	0	0	0
3	0	1	0	1	0	0	0	0
4	0	1	0	1	0	0	0	0

[5 rows x 376 columns]

[17]:
```
print('Feature types:')
x_train[cols].dtypes.value_counts()
```

```
Feature types:
```

[17]:
```
int64    376
dtype: int64
```

[18]:
```
from sklearn.decomposition import PCA
```

[19]:
```
n_comp = 12
pca = PCA(n_components=n_comp, random_state=420)
pca2_results_train = pca.fit_transform(x_train)
```

```
pca2_results_test = pca.transform(x_test)
```

[20]:
```
pca2_results_train
```

[20]:
```
array([[-49.08156207,  -4.90948084, -17.25085325, …,   1.6580127 ,
          0.93309992,   1.67820505],
       [-48.94680383,  -7.22674339, -13.7631947 , …,  -0.21430878,
          0.10908942,   0.44947029],
       [ 92.62761708,  31.9940341 , -26.17503456, …,  -0.62193667,
          2.92597561,  -0.5274524 ],
       …,
       [ 89.47970814,  20.44554421,  48.11999819, …,  -1.27197787,
         -0.28720258,   2.00793054],
       [ 96.97110845,  31.50977186,  49.20059282, …,   0.14366082,
         -0.97983972,   0.99197679],
       [-17.21024322, -14.22166025,  55.38091289, …,  -0.28903843,
         -0.31649851,   0.69134356]])
```

[21]:
```
pca2_results_test
```

[21]:
```
array([[ 9.22615149e+01,  3.29260839e+01, -3.01130736e+01, …,
        -4.11416844e-01,  3.62102894e+00, -1.20770031e+00],
       [-3.48622379e+01,  6.87132606e+00, -3.74760829e+01, …,
         6.09260099e-01, -6.95791780e-01, -4.24922762e-01],
       [ 4.36560426e+01, -5.05939489e+01, -6.10591086e+01, …,
        -3.20449914e-01,  2.60139386e+00, -1.53758244e+00],
       …,
       [-2.52437784e+01, -2.63794193e+01,  5.40742341e+01, …,
         6.03523571e-01,  2.61335990e-02,  3.66998572e-02],
       [ 4.53823778e+01, -6.38062446e+01,  3.58666036e+01, …,
        -9.15193005e-01, -6.72301188e-01,  5.15233353e-01],
       [-4.23807477e+01, -2.52862351e+01,  6.10815522e+01, …,
        -2.98845820e-01, -9.77125995e-01,  5.35472021e-02]])
```

[22]:
```
import xgboost as xgb
from sklearn.metrics import r2_score
```

[23]:
```
from sklearn.model_selection import train_test_split
```

[24]:
```
x_train, x_test, y_train, y_test = train_test_split(pca2_results_train,y_train,␣
↪test_size=0.2,random_state=4242)
```

[25]:
```
d_train = xgb.DMatrix(x_train, label=y_train)
d_valid = xgb.DMatrix(x_test, label=y_test)
d_test = xgb.DMatrix(pca2_results_test)
```

[26]:
```
params = {}
params['objective'] = 'reg:linear'
params['eta'] = 0.02
params['max_depth'] = 4
```

[27]:
```
def xgb_r2_score(preds, dtrain):
    labels = dtrain.get_label()
    return 'r2', r2_score(labels, preds)

watchlist = [(d_train, 'train'), (d_valid, 'valid')]
```

[28]:
```
clf = xgb.train(params, d_train,1000, watchlist,␣
↪early_stopping_rounds=50,feval=xgb_r2_score, maximize=True, verbose_eval=10)
```

```
[08:13:03] WARNING: C:/Users/Administrator/workspace/xgboost-
win64_release_1.5.1/src/objective/regression_obj.cu:188: reg:linear is now
deprecated in favor of reg:squarederror.
[0]    train-rmse:99.14838    train-r2:-58.35295    valid-rmse:98.26298
valid-r2:-67.63754
[10]   train-rmse:81.27657    train-r2:-38.88428    valid-rmse:80.36432
valid-r2:-44.91014
[20]   train-rmse:66.71610    train-r2:-25.87403    valid-rmse:65.77333
valid-r2:-29.75260
[30]   train-rmse:54.86957    train-r2:-17.17752    valid-rmse:53.88974
valid-r2:-19.64401
[40]   train-rmse:45.24491    train-r2:-11.35979    valid-rmse:44.21970
valid-r2:-12.89996
[50]   train-rmse:37.44730    train-r2:-7.46666     valid-rmse:36.37237
valid-r2:-8.40428
[60]   train-rmse:31.14750    train-r2:-4.85757     valid-rmse:30.01874
valid-r2:-5.40571
[70]   train-rmse:26.08661    train-r2:-3.10872     valid-rmse:24.90890
valid-r2:-3.41053
[80]   train-rmse:22.04639    train-r2:-1.93458     valid-rmse:20.83274
valid-r2:-2.08514
[90]   train-rmse:18.84412    train-r2:-1.14399     valid-rmse:17.60176
valid-r2:-1.20239
[100]  train-rmse:16.33254    train-r2:-0.61057     valid-rmse:15.08443
valid-r2:-0.61748
[110]  train-rmse:14.39915    train-r2:-0.25183     valid-rmse:13.14876
valid-r2:-0.22900
[120]  train-rmse:12.91704    train-r2:-0.00739     valid-rmse:11.68509
valid-r2:0.02939
[130]  train-rmse:11.80783    train-r2:0.15819      valid-rmse:10.61558
valid-r2:0.19893
[140]  train-rmse:10.98127    train-r2:0.27192      valid-rmse:9.84689
valid-r2:0.31074
```

```
[150]   train-rmse:10.38013     train-r2:0.34946        valid-rmse:9.31964
valid-r2:0.38258
[160]   train-rmse:9.92687      train-r2:0.40503        valid-rmse:8.95210
valid-r2:0.43032
[170]   train-rmse:9.59379      train-r2:0.44428        valid-rmse:8.70924
valid-r2:0.46081
[180]   train-rmse:9.34597      train-r2:0.47262        valid-rmse:8.54908
valid-r2:0.48046
[190]   train-rmse:9.16139      train-r2:0.49325        valid-rmse:8.44279
valid-r2:0.49330
[200]   train-rmse:9.02101      train-r2:0.50866        valid-rmse:8.37317
valid-r2:0.50162
[210]   train-rmse:8.91451      train-r2:0.52019        valid-rmse:8.33436
valid-r2:0.50623
[220]   train-rmse:8.84139      train-r2:0.52803        valid-rmse:8.30600
valid-r2:0.50958
[230]   train-rmse:8.77508      train-r2:0.53509        valid-rmse:8.29487
valid-r2:0.51089
[240]   train-rmse:8.72891      train-r2:0.53996        valid-rmse:8.28582
valid-r2:0.51196
[250]   train-rmse:8.68671      train-r2:0.54440        valid-rmse:8.28124
valid-r2:0.51250
[260]   train-rmse:8.65006      train-r2:0.54824        valid-rmse:8.27879
valid-r2:0.51279
[270]   train-rmse:8.62073      train-r2:0.55130        valid-rmse:8.27846
valid-r2:0.51283
[280]   train-rmse:8.59450      train-r2:0.55402        valid-rmse:8.27990
valid-r2:0.51266
[290]   train-rmse:8.56474      train-r2:0.55711        valid-rmse:8.27987
valid-r2:0.51266
[300]   train-rmse:8.54270      train-r2:0.55938        valid-rmse:8.28130
valid-r2:0.51250
[310]   train-rmse:8.51389      train-r2:0.56235        valid-rmse:8.28044
valid-r2:0.51260
[317]   train-rmse:8.49794      train-r2:0.56399        valid-rmse:8.28072
valid-r2:0.51256
```

[29]:
```
p_test = clf.predict(d_test)
```

[30]:
```
sub = pd.DataFrame()
sub['ID'] = id_test
sub['y'] = p_test
sub.to_csv('xgb.csv', index=False)
```

[31]:
```
sub.head()
```

[31]:

	ID	y
0	1	82.747520
1	2	97.118401
2	3	83.643776
3	4	77.565765
4	5	111.726906

[ ]: