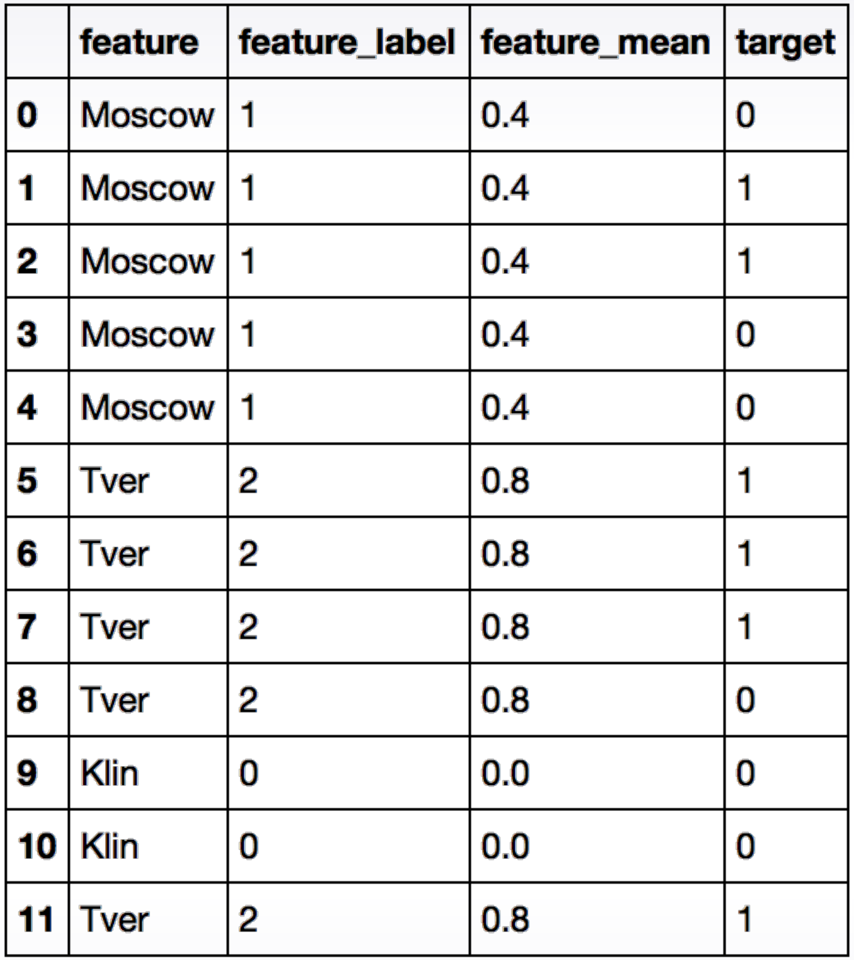
Using target to generate features

Some call it likelihood encoding, some target encoding

Simple example

- Categorical feature
  - some city
- Binary classification

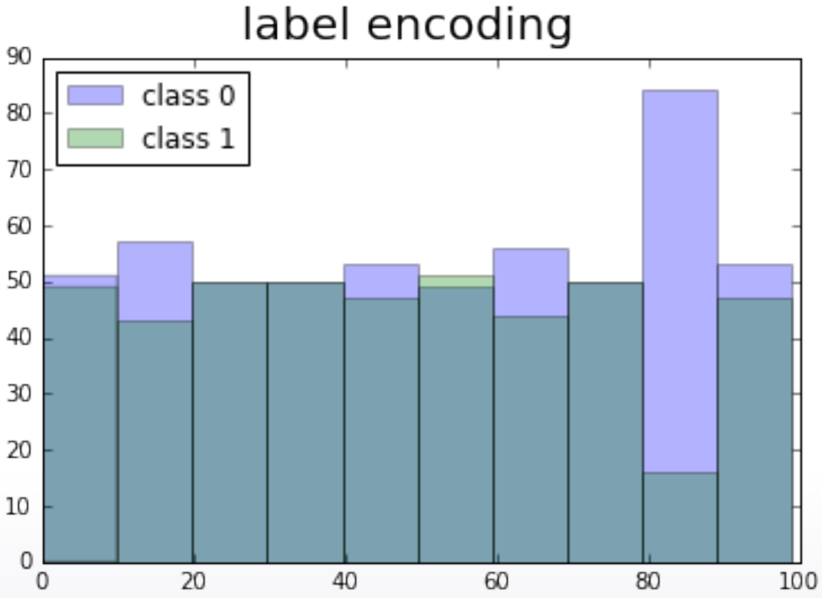
The most obvious way and what people usually use is label encoding. It's what we have in second column.

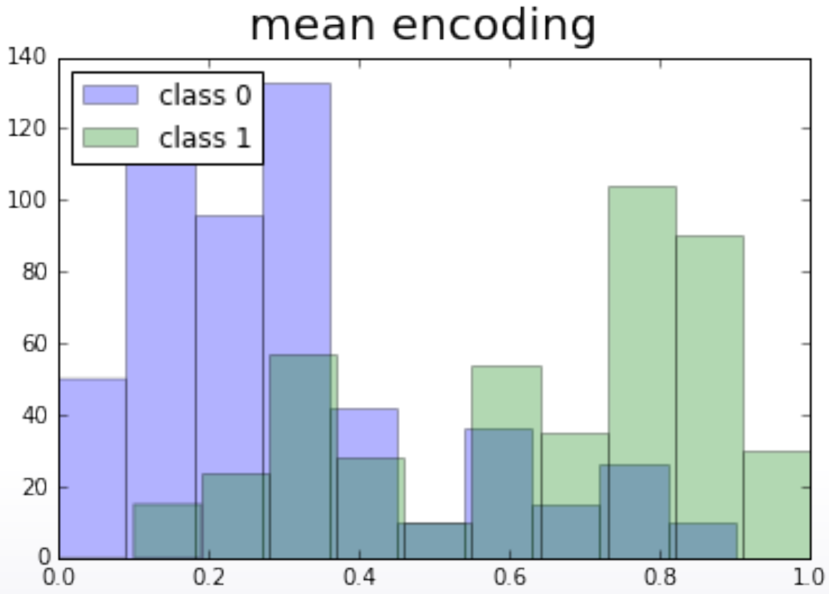


	feature	feature_label	feature_mean	target
0	Moscow	1	0.4	0
1	Moscow	1	0.4	1
2	Moscow	1	0.4	1
3	Moscow	1	0.4	0
4	Moscow	1	0.4	0
5	Tver	2	0.8	1
6	Tver	2	0.8	1
7	Tver	2	0.8	1
8	Tver	2	0.8	0
9	Klin	0	0.0	0
10	Klin	0	0.0	0
11	Tver	2	0.8	1

Why does it work?

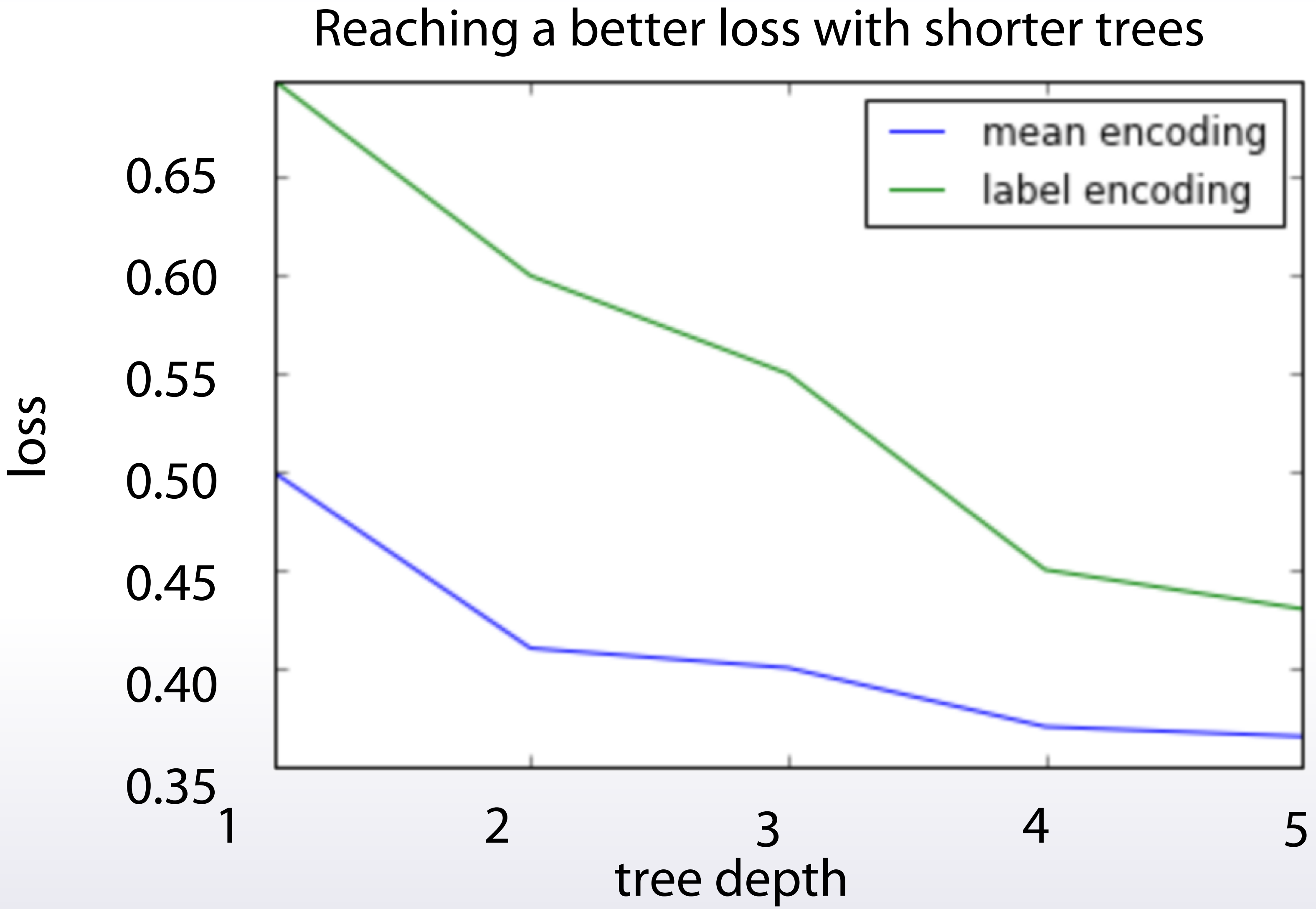
1. Label encoding gives random order. No correlation with target
2. Mean encoding helps to separate zeros from ones





Why does it work?

we can reach better loss with shorter trees. Cross validation loss might even look like this.
In general, the more complicated and non linear feature target dependency, the more effective is mean encoding.



most popular and effective way to solve machine learning problem? Is grading using trees, LGBM. One of the few downsides is an inability to handle high cardinality categorical variables.

Trees have limited depth, with mean encoding, we can compensate it, we can reach better loss with shorter trees. Cross validation loss might even look like this.

What will you learn?

- ✓ Construct encodings

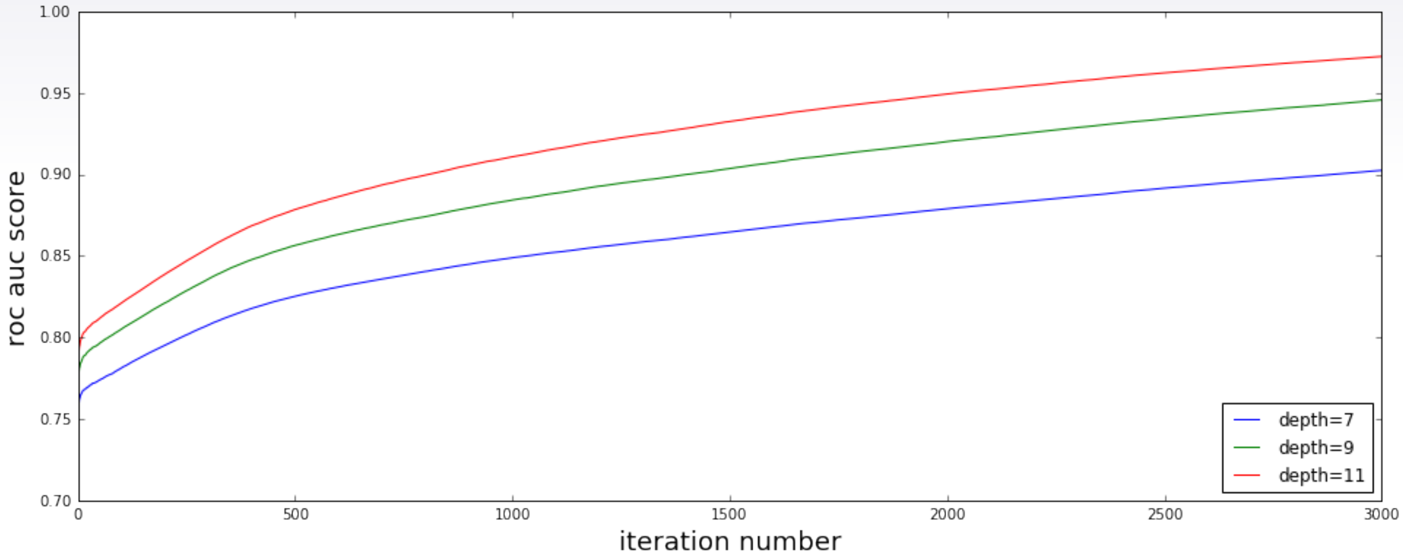
Despite the simplicity of the idea, you need to be very careful with validation

- ✓ Correctly validate them
- ✓ Extend them

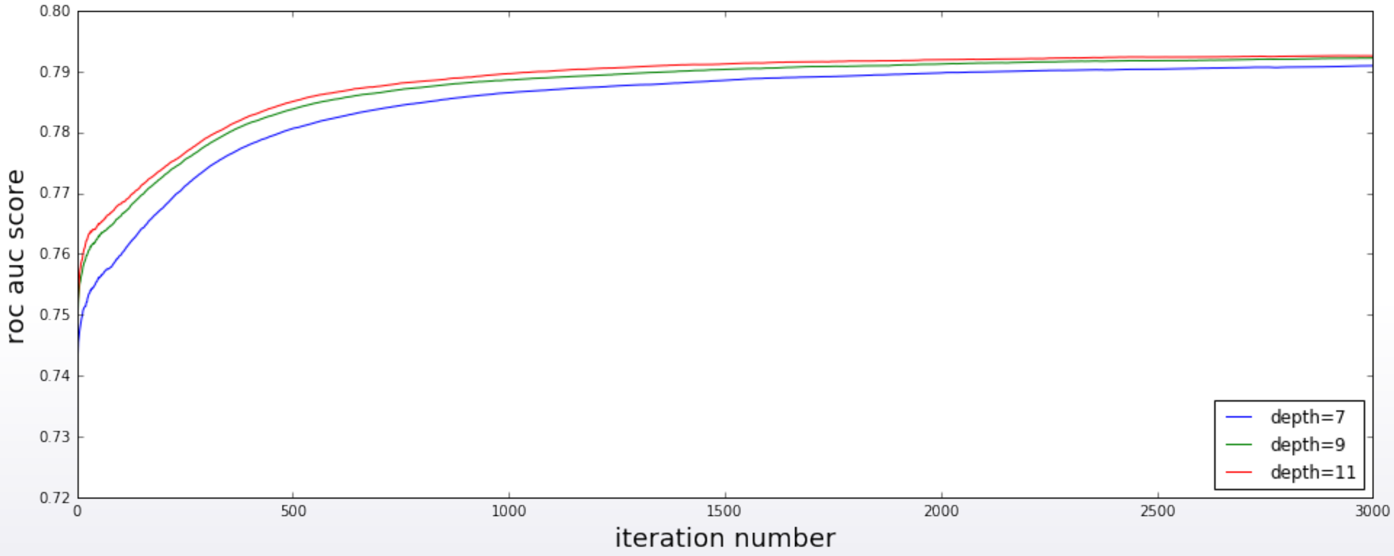
The last, but not least, are extensions. There are countless possibilities to derive new features from target variable. Sometimes, they produce significant improvement for your models.

Indicators of usefulness

Train



Validation



Ways to use target variable

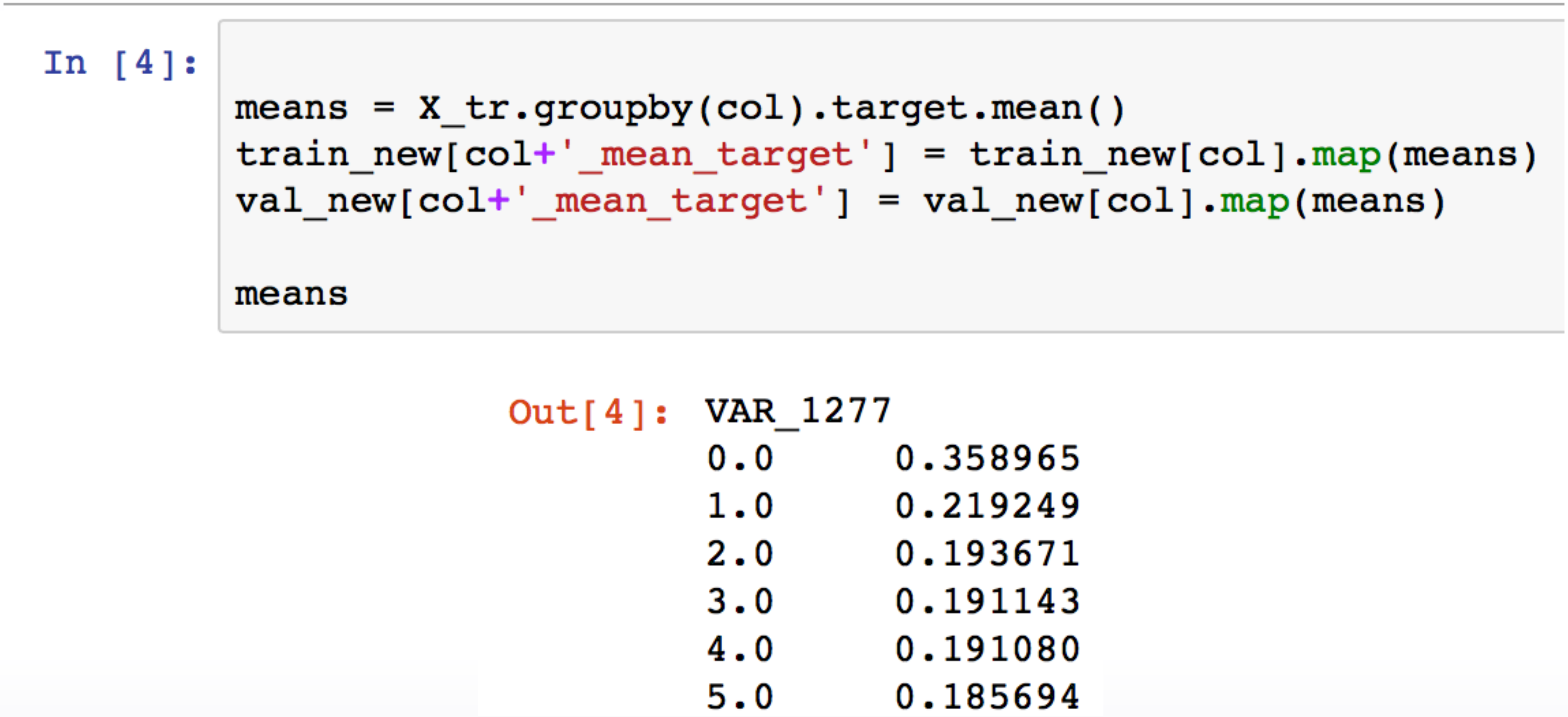
Goods - number of ones in a group,

Bads - number of zeros

- $Likelihood = \frac{Goods}{Goods+Bads} = mean(target)$
- $Weight\ of\ Evidence = \ln\left(\frac{Goods}{Bads}\right) * 100$
- $Count = Goods = sum(target)$
- $Diff = Goods - Bads$

Springleaf example

先split dataset 然后mean encoding



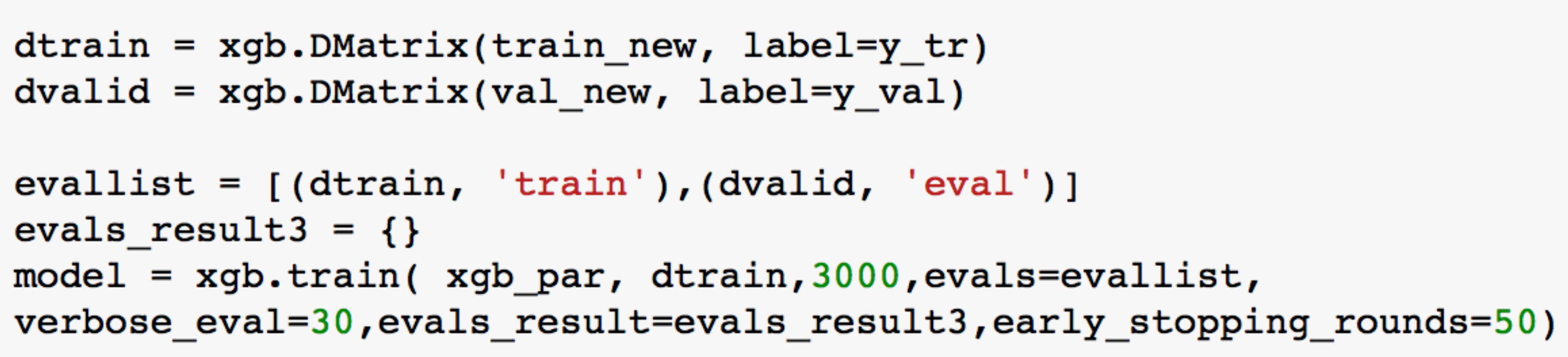
In [4]:

```
means = X_tr.groupby(col).target.mean()
train_new[col+'_mean_target'] = train_new[col].map(means)
val_new[col+'_mean_target'] = val_new[col].map(means)

means
```

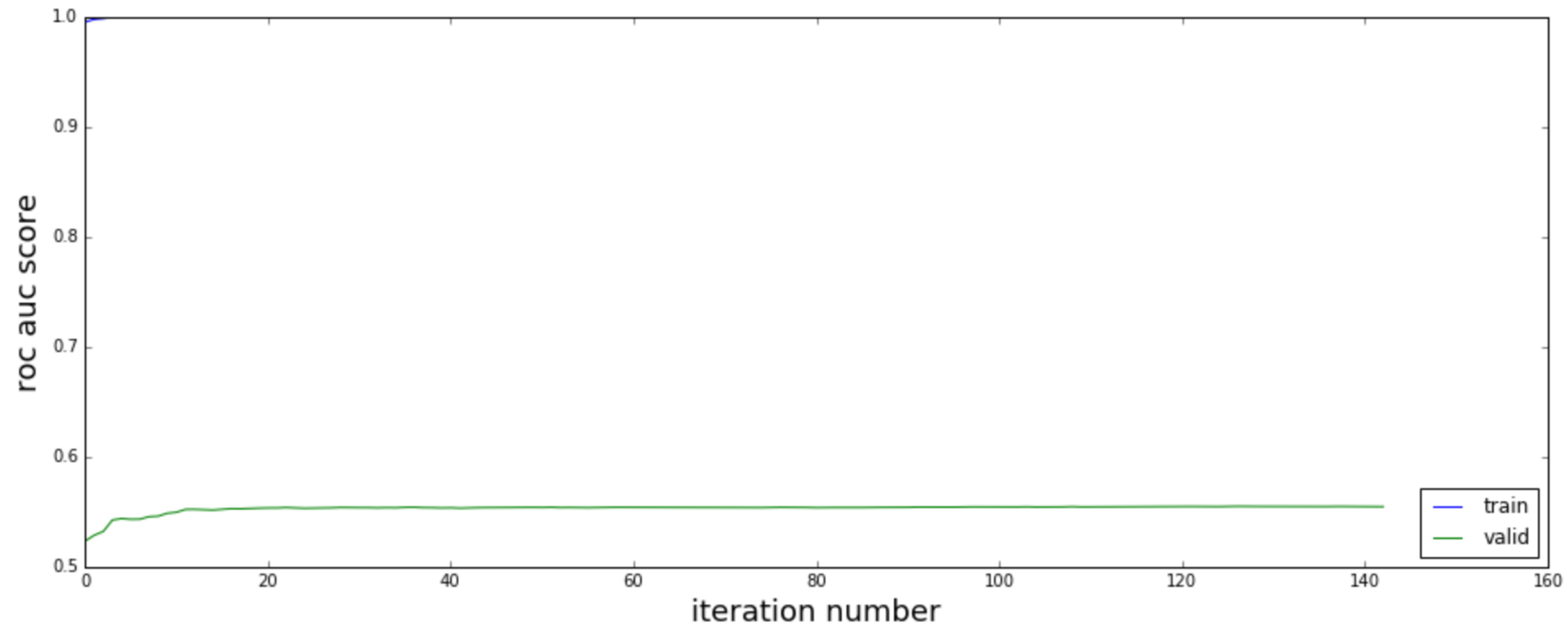
Out[4]:

```
VAR_1277
0.0       0.358965
1.0       0.219249
2.0       0.193671
3.0       0.191143
4.0       0.191080
5.0       0.185694
```

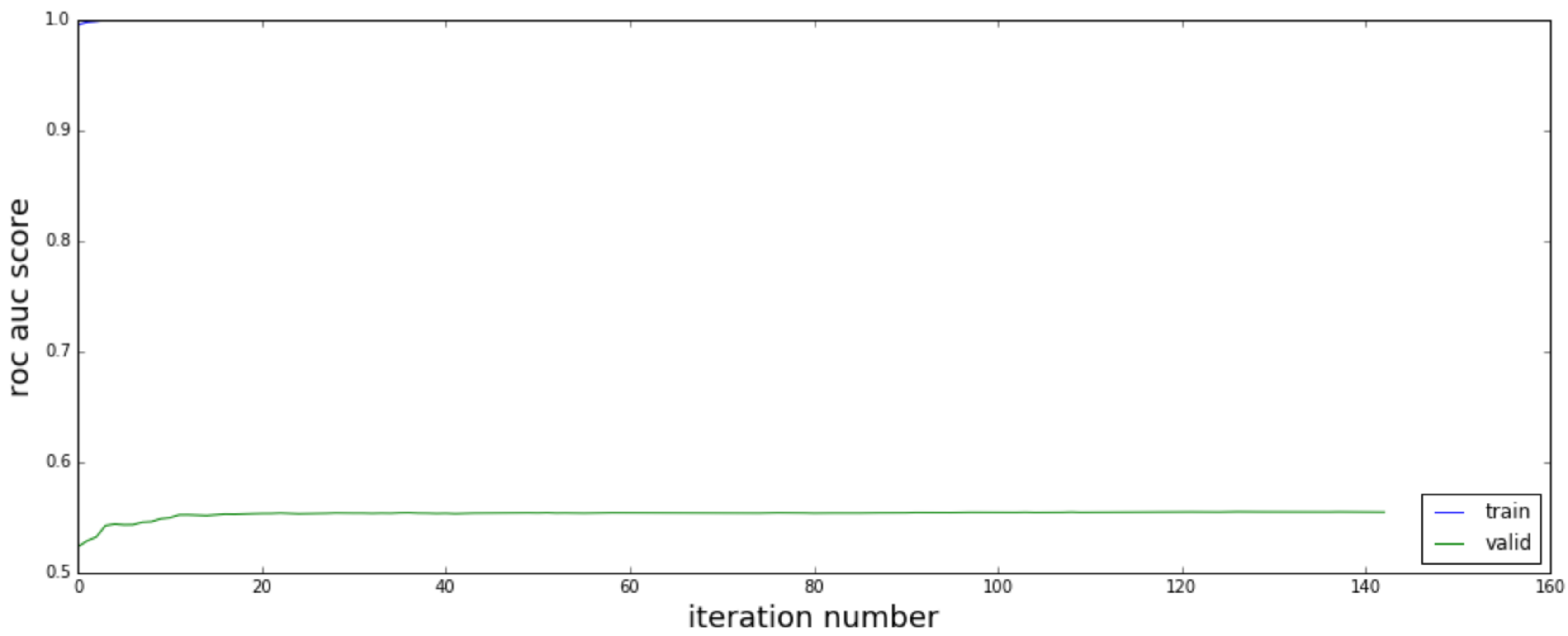
Springleaf example



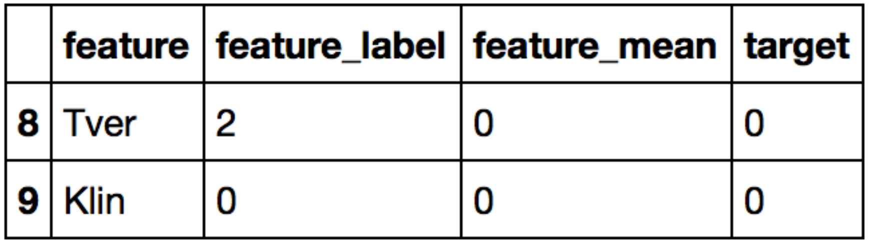
```
dtrain = xgb.DMatrix(train_new, label=y_tr)
dvalid = xgb.DMatrix(val_new, label=y_val)

evallist = [(dtrain, 'train'),(dvalid, 'eval')]
evals_result3 = {}
model = xgb.train( xgb_par, dtrain,3000,evals=evallist,
verbose_eval=30,evals_result=evals_result3,early_stopping_rounds=50)
```



Overfit

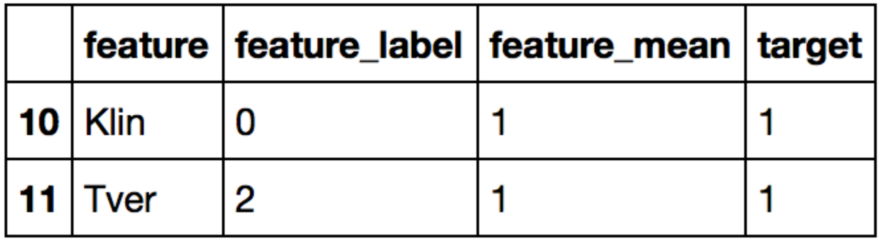


Train



	feature	feature_label	feature_mean	target
8	Tver	2	0	0
9	Klin	0	0	0

Validation



	feature	feature_label	feature_mean	target
10	Klin	0	1	1
11	Tver	2	1	1