

<div>Submission</div> <div>✓ Ran successfully</div> <div>Submitted by Denis Larionov a year ago</div>	<div>Public Score</div> <div>0.80382</div>
---	--

Ridiculous, but best score on the test set is reached without feature engineering and tuning model parameters. It looks random.

Loading data

In [1]:

```
import numpy as np
import pandas as pd

from catboost import CatBoostClassifier, Pool, cv

import hyperopt
```

In [2]:

```
train = pd.read_csv('../input/train.csv')
test = pd.read_csv('../input/test.csv')

train_size = train.shape[0] # 891
test_size = test.shape[0] # 418

data = pd.concat([train, test])
```

```
/opt/conda/lib/python3.6/site-packages/ipykernel_launcher.py:7: FutureWarning: Sorting because non-concatenation axis is not aligned. A future version of pandas will change to not sort by default.
```

To accept the future behavior, pass 'sort=True'.

To retain the current behavior and silence the warning, pass sort=False

```
import sys
```

Feature engineering

In [3]:

```
data['Title'] = data['Name'].str.extract('([A-Za-z]+)\.',  
    , expand=False)
```

For each title calculat mean age and fill nan with it.

In [4]:

```
age_ref = data.groupby('Title').Age.mean()  
data['Age'] = data.apply(lambda r: r.Age if pd.notnull(r  
.Age) else age_ref[r.Title] , axis=1)  
del age_ref
```

One missing Fare. Impute it with mean by Pclass=3 and Embarked=S

In [5]:

```
data.loc[(data.PassengerId==1044, 'Fare')] = 14.43
```

In [6]:

```
data['Embarked'] = data['Embarked'].fillna('S')  
data['Cabin'] = data['Cabin'].fillna('Undefined')
```

Training

In [7]:

```
cols = [  
    'Pclass',  
    'Name',  
    'Sex',  
    'Age',  
    'SibSp',  
    'Parch',  
    'Ticket',  
    'Fare',  
    'Cabin',  
    'Embarked'  
]  
X_train = data[:train_size][cols]  
Y_train = data[:train_size]['Survived'].astype(int)
```

```
X_test = data[train_size:][cols]

categorical_features_indices = [0,1,2,6,8,9]
X_train.head()
```

Out[7]:

	Pclass	Name	Sex	Age	SibSp	Parch	Ticket
0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 2117
1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17
2	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON 3101
3	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	1138



Titanic catboost

Python notebook using data from [Titanic: Machine Learning from Disaster](#) · 1,221 views · beginner, classification, feature engineering



6

Fork

6



Version 36

36 commits

Notebook

Data

Output

Log

Comments

In [8]:

```
train_pool = Pool(X_train, Y_train, cat_features=categorical_features_indices)
```

Tune hyperparameters with hyperopt
(<https://github.com/hyperopt/hyperopt>) and cross-validation

In [9]:

```
'''
def hyperopt_objective(params):
    model = CatBoostClassifier(
        l2_leaf_reg=int(params['l2_leaf_reg']),
        learning_rate=params['learning_rate'],
        depth=params['depth'],
        iterations=500,
        eval_metric='Accuracy',
```

```

        od_type='Iter',
        od_wait=40,
        random_seed=42,
        logging_level='Silent',
        allow_writing_files=False
    )

    cv_data = cv(
        train_pool,
        model.get_params()
    )
    best_accuracy = np.max(cv_data['test-Accuracy-mean'])

    print(params, best_accuracy)
    return 1 - best_accuracy # as hyperopt minimises

params_space = {
    'l2_leaf_reg': hyperopt.hp.qloguniform('l2_leaf_reg',
0, 2, 1),
    'learning_rate': hyperopt.hp.uniform('learning_rate',
1e-3, 5e-1),
    'depth': hyperopt.hp.choice('depth', [3,4,5,6,8]),
}

```



Notebook



Data



Output



Log



Comments

```

best = hyperopt.fmin(
    hyperopt_objective,
    space=params_space,
    algo=hyperopt.tpe.suggest,
    max_evals=50,
    trials=trials
)

print(best)
'''

```

Out[9]:

```

"\ndef hyperopt_objective(params):\n    mod
el = CatBoostClassifier(\n        l2_leaf_r
eg=int(params['l2_leaf_reg']),\n        lea
rning_rate=params['learning_rate'],\n
depth=params['depth'],\n        iterations=
500,\n        eval_metric='Accuracy',\n
od_type='Iter',\n        od_wait=40,\n
random_seed=42,\n        logging_level='Sil
ent',\n        allow_writing_files=False\n
)\n    \n    cv_data = cv(\n        train_p
ool,\n        model.get_params()\n    )\n
best_accuracy = np.max(cv_data['test-Accura
cy-mean'])\n    \n    print(params, bes

```

```
t_accuracy)\n    return 1 - best_accuracy #
as hyperopt minimises\n\nparams_space = {\n
'l2_leaf_reg': hyperopt.hp.qloguniform('l2_
leaf_reg', 0, 2, 1),\n    'learning_rate':
hyperopt.hp.uniform('learning_rate', 1e-3,
5e-1),\n    'depth': hyperopt.hp.choice('de
pth', [3,4,5,6,8]),\n}\n\ntrials = hyperop
t.Trials()\n\nbest = hyperopt.fmin(\n    hy
peropt_objective,\n    space=params_spac
e,\n    algo=hyperopt.tpe.suggest,\n    max
_evals=50,\n    trials=trials\n)\n\nprint(b
est)\n"
```

In [10]:

```
#best = {'depth': 6, 'l2_leaf_reg': 1.0, 'learning_rate':
0.07395682681736576}
```

```
model = CatBoostClassifier(
    #l2_leaf_reg=int(best['l2_leaf_reg']),
    #learning_rate=best['learning_rate'],
    #depth=best['depth'],
    depth=3,
    iterations=300,
    eval_metric='Accuracy',
    #od_type='Iter',
    #od_wait=40,
    random_seed=42,
    logging_level='Silent',
    allow_writing_files=False
)

cv_data = cv(
    train_pool,
    model.get_params(),
    fold_count=5
)

print('Best validation accuracy score: {:.2f}±{:.2f} on
step {}'.format(
    np.max(cv_data['test-Accuracy-mean']),
    cv_data['test-Accuracy-std'][cv_data['test-Accuracy-
mean'].idxmax(axis=0)],
    cv_data['test-Accuracy-mean'].idxmax(axis=0)
))
print('Precise validation accuracy score: {}'.format(np.
max(cv_data['test-Accuracy-mean'])))

model.fit(train_pool);
model.score(X_train, Y_train)
```

Best validation accuracy score: 0.84±0.03 on step 106
 Precise validation accuracy score: 0.8406440273680247

Out[10]:

0.92143658810325479

This kernel has been released under the [Apache 2.0](#) open source license.

In [11]:

```
feature_importances = model.get_feature_importance(train_pool)
```

Did you find this Kernel useful?

Show your appreciation with an upvote

6



Data

Data Sources

▼ 🏆 Titanic: Machi...

... 418 x 2

t 418 x 11

t 891 x 12



Titanic: Machine Learning from Disaster

Start here! Predict survival on the Titanic and get familiar with ML basics

Last Updated: 7 years ago

About this Competition

Overview

The data has been split into two groups:

- training set (train.csv)
- test set (test.csv)

The training set should be used to build your machine learning models. For the training set, we provide the outcome (also known as the "ground truth") for each passenger. Your model will be based on "features" like passengers' gender and class. You can also use feature engineering to create new features.

The test set should be used to see how well your model performs on unseen data. For the test set, we do not provide the ground truth for each passenger. It is your job to predict these outcomes. For each passenger in the test set, use the model you trained to predict whether or not they survived the sinking of the Titanic.

We also include **gender_submission.csv**, a set of predictions that assume all and only female

passengers survive, as an example of what a submission file should look like.

Output Files

New Dataset

New Kernel

Download All



Output Files

submission.csv

About this file

This file was created from a Kernel, it does not have a description.

submission.csv



1	PassengerId	Survived
2	892	0
3	893	0
4	894	0
5	895	0
6	896	1
7	897	0
8	898	1
9	899	0
10	900	1
11	901	0
12	902	0
13	903	0
14	904	1
15	905	0
16	906	1
17	907	1
18	908	0
19	909	0
20	910	1
21	911	1
22	912	0
23	913	0
24	914	1
25	915	0
26	916	1
27	917	0
28	918	1

29	919	0
30	920	0
31	921	0

Run Info


Succeeded	True	Run Time	185 seconds
Exit Code	0	Queue Time	0 seconds
Docker Image Name	kaggle/python (Dockerfile)		
Timeout Exceeded	False	Output Size	0
		Used All Space	False
Failure Message			

Log

[Download Log](#)

Time	Line #	Log Message
3.4s	1	[NbConvertApp] Converting notebook script.ipynb to html
3.4s	2	[NbConvertApp] Executing notebook with kernel: python3
184.4s	3	[NbConvertApp] Writing 276116 bytes to __results__.html
184.4s	4	
184.4s	6	Complete. Exited with code 0.

Comments (0)



Click here to enter a comment...

Similar Kernels

