```
!pip install catboost
```

```
Collecting catboost
  Downloading catboost-1.0.5-cp37-none-manylinux1_x86_64.whl (76.6 MB)
     |████████████████████████████████| 76.6 MB 1.4 MB/s
Requirement already satisfied: graphviz in /usr/local/lib/python3.7/dist-packa
Requirement already satisfied: scipy in /usr/local/lib/python3.7/dist-packages
Requirement already satisfied: plotly in /usr/local/lib/python3.7/dist-package
Requirement already satisfied: pandas>=0.24.0 in /usr/local/lib/python3.7/dist
Requirement already satisfied: six in /usr/local/lib/python3.7/dist-packages (
Requirement already satisfied: matplotlib in /usr/local/lib/python3.7/dist-pac
Requirement already satisfied: numpy>=1.16.0 in /usr/local/lib/python3.7/dist-
Requirement already satisfied: python-dateutil>=2.7.3 in /usr/local/lib/python
Requirement already satisfied: pytz>=2017.3 in /usr/local/lib/python3.7/dist-p
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.7/dist-p
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in /us
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.7/d
Requirement already satisfied: typing-extensions in /usr/local/lib/python3.7/d
Requirement already satisfied: tenacity>=6.2.0 in /usr/local/lib/python3.7/dis
Installing collected packages: catboost
Successfully installed catboost-1.0.5
```

Решение команды DataKit в рамках хакатона 13-15 мая. Кейс 9. Предсказание подозрительных операций по банковским картам. Датасет https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud?resource=download

```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split   #to divide dataset in train
#Import scikit-learn metrics module for accuracy calculation
from catboost import CatBoostClassifier
from sklearn import metrics
```

```
RS = 42  #we will fix random state
```

```
# подключаем гугл диск на котором данные
from google.colab import drive
drive.mount ('/content/gdrive', force_remount = True)
```

```
Mounted at /content/gdrive
```

```
!cp /content/gdrive/'My Drive'/2022projects/Hack_Rostov_may13_15/archive.zip .
```

```
!ls
```

```
archive.zip  creditcard.csv  gdrive  sample_data
```

```
!unzip archive.zip
```

```
Archive:  archive.zip
  inflating: creditcard.csv
```

```
data_df = pd.read_csv('creditcard.csv')
data_df.head(2)
```

	Time	V1	V2	V3	V4	V5	V6	V7	
0	0.0	-1.359807	-0.072781	2.536347	1.378155	-0.338321	0.462388	0.239599	0.0986
1	0.0	1.191857	0.266151	0.166480	0.448154	0.060018	-0.082361	-0.078803	0.0851

2 rows × 31 columns

```
data_df.describe().T
```

	count	mean	std	min	25%	50%
Time	284807.0	9.481386e+04	47488.145955	0.000000	54201.500000	84692.000000
V1	284807.0	3.918649e-15	1.958696	-56.407510	-0.920373	0.018109
V2	284807.0	5.682686e-16	1.651309	-72.715728	-0.598550	0.065486
V3	284807.0	-8.761736e-15	1.516255	-48.325589	-0.890365	0.179846
V4	284807.0	2.811118e-15	1.415869	-5.683171	-0.848640	-0.019847
V5	284807.0	-1.552103e-15	1.380247	-113.743307	-0.691597	-0.054336
V6	284807.0	2.040130e-15	1.332271	-26.160506	-0.768296	-0.274187
V7	284807.0	-1.698953e-15	1.237094	-43.557242	-0.554076	0.040103
V8	284807.0	-1.893285e-16	1.194353	-73.216718	-0.208630	0.022358

```
train_df = data_df[:280_000]
test_df = data_df[280_000:]
```

```
data_df.shape
```

```
(284807, 31)
```

```
train_df.columns
```

```
Index(['Time', 'V1', 'V2', 'V3', 'V4', 'V5', 'V6', 'V7', 'V8', 'V9', 'V10',
       'V11', 'V12', 'V13', 'V14', 'V15', 'V16', 'V17', 'V18', 'V19', 'V20',
       'V21', 'V22', 'V23', 'V24', 'V25', 'V26', 'V27', 'V28', 'Amount',
       'Class'],
      dtype='object')
```

```
X_features = ['V1', 'V2', 'V3', 'V4', 'V5', 'V6', 'V7', 'V8', 'V9', 'V10',
       'V11', 'V12', 'V13', 'V14', 'V15', 'V16', 'V17', 'V18', 'V19', 'V20',
       'V21', 'V22', 'V23', 'V24', 'V25', 'V26', 'V27', 'V28', 'Amount']
```

V20	284807.0	5.126845e-16	0.770925	-54.497720	-0.211721	-0.062481

```
#DIVIDE DATA TO TRAIN AND TEST
X_train, X_val, y_train, y_val = train_test_split(train_df[X_features],
                                                  train_df['Cla
                                                  shuffle=True,
                                                  test_size=0.3
```

```
clf = CatBoostClassifier(
    iterations=100,
    depth = 11,
    learning_rate=0.1,
```

```
    loss_function='CrossEntropy',
    random_seed = RS
)


#Train the model using the training sets y_pred=clf.predict(X_test)
clf.fit(X_train,y_train)
```

```
0:      learn: 0.3929621        total: 296ms    remaining: 29.4s
1:      learn: 0.2207760        total: 592ms    remaining: 29s
2:      learn: 0.1259318        total: 902ms    remaining: 29.2s
3:      learn: 0.0723031        total: 1.19s    remaining: 28.6s
4:      learn: 0.0453382        total: 1.5s     remaining: 28.6s
5:      learn: 0.0290747        total: 1.81s    remaining: 28.3s
6:      learn: 0.0197568        total: 2.1s     remaining: 27.9s
7:      learn: 0.0141205        total: 2.41s    remaining: 27.8s
8:      learn: 0.0105698        total: 2.7s     remaining: 27.3s
9:      learn: 0.0080658        total: 3s       remaining: 27s
10:     learn: 0.0064716        total: 3.29s    remaining: 26.6s
11:     learn: 0.0053803        total: 3.58s    remaining: 26.3s
12:     learn: 0.0045213        total: 3.88s    remaining: 26s
13:     learn: 0.0039569        total: 4.17s    remaining: 25.6s
14:     learn: 0.0035453        total: 4.46s    remaining: 25.3s
15:     learn: 0.0032120        total: 4.77s    remaining: 25s
16:     learn: 0.0029557        total: 5.06s    remaining: 24.7s
17:     learn: 0.0027603        total: 5.35s    remaining: 24.4s
18:     learn: 0.0025987        total: 5.64s    remaining: 24s
19:     learn: 0.0024812        total: 5.93s    remaining: 23.7s
20:     learn: 0.0023710        total: 6.21s    remaining: 23.4s
21:     learn: 0.0022835        total: 6.5s     remaining: 23s
22:     learn: 0.0022218        total: 6.77s    remaining: 22.7s
23:     learn: 0.0021457        total: 7.07s    remaining: 22.4s
24:     learn: 0.0020800        total: 7.36s    remaining: 22.1s
25:     learn: 0.0020408        total: 7.64s    remaining: 21.7s
26:     learn: 0.0019986        total: 7.94s    remaining: 21.5s
27:     learn: 0.0019612        total: 8.22s    remaining: 21.1s
28:     learn: 0.0019198        total: 8.52s    remaining: 20.9s
29:     learn: 0.0018967        total: 8.8s     remaining: 20.5s
30:     learn: 0.0018544        total: 9.1s     remaining: 20.3s
31:     learn: 0.0018242        total: 9.39s    remaining: 20s
32:     learn: 0.0017926        total: 9.68s    remaining: 19.7s
33:     learn: 0.0017657        total: 9.96s    remaining: 19.3s
34:     learn: 0.0017383        total: 10.3s    remaining: 19.1s
35:     learn: 0.0017232        total: 10.6s    remaining: 18.8s
36:     learn: 0.0017010        total: 10.8s    remaining: 18.5s
37:     learn: 0.0016887        total: 11.1s    remaining: 18.2s
38:     learn: 0.0016678        total: 11.4s    remaining: 17.9s
39:     learn: 0.0016498        total: 11.7s    remaining: 17.5s
40:     learn: 0.0016215        total: 12s      remaining: 17.2s
41:     learn: 0.0015993        total: 12.3s    remaining: 16.9s
42:     learn: 0.0015846        total: 12.6s    remaining: 16.7s
43:     learn: 0.0015461        total: 12.9s    remaining: 16.4s
44:     learn: 0.0015277        total: 13.2s    remaining: 16.1s
45:     learn: 0.0015052        total: 13.5s    remaining: 15.8s
46:     learn: 0.0014868        total: 13.7s    remaining: 15.5s
47:     learn: 0.0014792        total: 14s      remaining: 15.2s
48:     learn: 0.0014640        total: 14.3s    remaining: 14.9s
49:     learn: 0.0014383        total: 14.6s    remaining: 14.6s
50:     learn: 0.0014239        total: 14.9s    remaining: 14.3s
51:     learn: 0.0013807        total: 15.2s    remaining: 14s
```

```
52:     learn: 0.0013509        total: 15.5s    remaining: 13.7s
53:     learn: 0.0013284        total: 15.8s    remaining: 13.5s
54:     learn: 0.0013149        total: 16.1s    remaining: 13.2s
55:     learn: 0.0013054        total: 16.4s    remaining: 12.9s
56:     learn: 0.0012892        total: 16.7s    remaining: 12.6s
57:     learn: 0.0012739        total: 17s      remaining: 12.3s
```

```
y_pred=clf.predict(X_val)
y_pred[:10]
```

```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0])
```

```
feature_importance = sorted(zip(map(lambda x: round(x, 4),
                                    clf.feature_importances_),
                                X_features),
             reverse=True)
```

```
for i in range(len(feature_importance)):
  print(feature_importance[i])
```

```
(7.5517, 'V26')
(6.9818, 'V1')
(6.9344, 'Amount')
(5.3564, 'V15')
(4.7185, 'V4')
(4.6177, 'V28')
(4.4398, 'V12')
(4.4149, 'V13')
(4.3391, 'V17')
(3.6747, 'V22')
(3.5708, 'V21')
(3.4308, 'V14')
(3.4221, 'V27')
(3.1338, 'V8')
(3.1163, 'V24')
(2.8686, 'V6')
(2.8682, 'V9')
(2.8324, 'V10')
(2.5361, 'V25')
(2.4976, 'V16')
(2.4905, 'V18')
(2.4647, 'V11')
(2.128, 'V5')
(1.9781, 'V7')
(1.926, 'V20')
(1.7523, 'V3')
(1.6969, 'V19')
(1.2703, 'V23')
(0.9875, 'V2')
```

```
#X_val[1917:1925].to_csv("test.csv")
```

```
clf.score(X_val,y_val)
```

```
0.9996666666666667
```

```
y_t = clf.predict(test_df[X_features])
clf.score(y_t,test_df['Class'])
```

```
0.9991678801747451
```

```
metrics.f1_score(y_val,y_pred)
```

```
0.8870967741935484
```

```
#построим матрицу сопряженности confusion matrix
target_names = ['fraud operation 1', 'normal operation 0']
report = metrics.classification_report(y_val, y_pred,target_names=target_names)
report.split('\n')
print(report[:170])
```

```
                       precision    recall  f1-score   support

      fraud operation 1       1.00      1.00      1.00     83864
     normal operation 0       0.98      0.81      0.89
```

ошибки первого рода - пропуск цели

ошибки второго рода - ложное срабатывание

```
clf.save_model('clf_cb03.cbm',
               format='cbm')
```

```
!ls
```

```
archive.zip    clf_cb03.cbm    gdrive       test.csv
catboost_info  creditcard.csv  sample_data
```

```
#load binary file, we will use in web application
from google.colab import files
files.download('clf_cb03.cbm')
```

✓ 0 сек. выполнено в 09:18