In [1]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
```

In [2]:
```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

In [3]:
```
train = pd.read_csv("/content/drive/My Drive/DSN AI Bootcamp Qualification Hackat
test = pd.read_csv("/content/drive/My Drive/DSN AI Bootcamp Qualification Hackath
sample = pd.read_csv("/content/drive/My Drive/DSN AI Bootcamp Qualification Hacka
```

In [4]:
```
print(train.shape)
print(test.shape)
print(sample.shape)
```

```
(56000, 52)
(24000, 51)
(24000, 2)
```

In [5]:
```
submit = test[['Applicant_ID']]
```

In [6]:
```
train.fillna(-1, inplace = True)
test.fillna(-1, inplace = True)
```

In [7]:
```
train.replace(np.inf, -1, inplace = True)
test.replace(np.inf, -1, inplace = True)
```

In [8]:
```
train.drop('Applicant_ID', axis = 1, inplace= True)
test.drop('Applicant_ID', axis = 1, inplace= True)
```

In [9]:
```
train[['form_field47','default_status']] = train[['form_field47','default_status'
test['form_field47'] = test['form_field47'].astype('category')
```

In [10]:
```
train['form_field47'] = train['form_field47'].cat.codes
train['default_status'] = train['default_status'].cat.codes
test['form_field47'] = test['form_field47'].cat.codes
```

In [11]:
```
X = train.drop('default_status', axis = 1)
y = train['default_status']
```

In [12]:
```
!pip install catboost
!pip install lightgbm
```

```
Collecting catboost
  Downloading https://files.pythonhosted.org/packages/90/86/c3dcb600b4f9e7584ed
90ea9d30a717fb5c0111574675f442c3e7bc19535/catboost-0.24.1-cp36-none-manylinux1_
x86_64.whl (https://files.pythonhosted.org/packages/90/86/c3dcb600b4f9e7584ed90
ea9d30a717fb5c0111574675f442c3e7bc19535/catboost-0.24.1-cp36-none-manylinux1_x8
6_64.whl) (66.1MB)
     |████████████████████████████████| 66.1MB 54kB/s
Requirement already satisfied: numpy>=1.16.0 in /usr/local/lib/python3.6/dist-p
ackages (from catboost) (1.18.5)
Requirement already satisfied: scipy in /usr/local/lib/python3.6/dist-packages
 (from catboost) (1.4.1)
Requirement already satisfied: pandas>=0.24.0 in /usr/local/lib/python3.6/dist-
packages (from catboost) (1.1.2)
Requirement already satisfied: graphviz in /usr/local/lib/python3.6/dist-packag
es (from catboost) (0.10.1)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.6/dist-pack
ages (from catboost) (3.2.2)
Requirement already satisfied: six in /usr/local/lib/python3.6/dist-packages (f
rom catboost) (1.15.0)
Requirement already satisfied: plotly in /usr/local/lib/python3.6/dist-packages
(from catboost) (4.4.1)
Requirement already satisfied: python-dateutil>=2.7.3 in /usr/local/lib/python
3.6/dist-packages (from pandas>=0.24.0->catboost) (2.8.1)
Requirement already satisfied: pytz>=2017.2 in /usr/local/lib/python3.6/dist-pa
ckages (from pandas>=0.24.0->catboost) (2018.9)
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.6/di
st-packages (from matplotlib->catboost) (1.2.0)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in /us
r/local/lib/python3.6/dist-packages (from matplotlib->catboost) (2.4.7)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.6/dist-pa
ckages (from matplotlib->catboost) (0.10.0)
Requirement already satisfied: retrying>=1.3.3 in /usr/local/lib/python3.6/dist
-packages (from plotly->catboost) (1.3.3)
Installing collected packages: catboost
Successfully installed catboost-0.24.1
Requirement already satisfied: lightgbm in /usr/local/lib/python3.6/dist-packag
es (2.2.3)
Requirement already satisfied: numpy in /usr/local/lib/python3.6/dist-packages
 (from lightgbm) (1.18.5)
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.6/dist-pa
ckages (from lightgbm) (0.22.2.post1)
Requirement already satisfied: scipy in /usr/local/lib/python3.6/dist-packages
 (from lightgbm) (1.4.1)
Requirement already satisfied: joblib>=0.11 in /usr/local/lib/python3.6/dist-pa
ckages (from scikit-learn->lightgbm) (0.16.0)
```

In [13]:
```
from sklearn.model_selection import StratifiedKFold
from sklearn.metrics import accuracy_score, roc_auc_score
from catboost import CatBoostClassifier
from lightgbm import LGBMClassifier
```

In [14]:
```
kf = StratifiedKFold(n_splits = 20, random_state = 1)
```

```
/usr/local/lib/python3.6/dist-packages/sklearn/model_selection/_split.py:296: F
utureWarning: Setting a random_state has no effect since shuffle is False. This
will raise an error in 0.24. You should leave random_state to its default (Non
e), or set shuffle=True.
  FutureWarning
```

In [ ]:
```
# # INITIALIZE THE MODEL
cat4 = CatBoostClassifier(learning_rate = 0.01, max_depth = 8, n_estimators = 500
                          eval_metric = 'AUC',bootstrap_type = 'Bayesian',random_s
```

In [ ]:
```
scores40 = []
scores41 = []
for fold,(tr_in,te_in) in enumerate(kf.split(X, y)):
    print(f"================================Fold{fold}============================
    X_train,X_test = X.iloc[tr_in],X.iloc[te_in]
    y_train,y_test = y.iloc[tr_in],y.iloc[te_in]
    cat4.fit(X_train,y_train,eval_set=[(X_train,y_train),(X_test,y_test)],early_s
    y_pred = cat4.predict(X_test)
    scores40.append(accuracy_score(y_test,y_pred))
    scores41.append(roc_auc_score(y_test,y_pred))
```

```
================================Fold0====================================
0:      test: 0.7684223 test1: 0.7533164        best: 0.7533164 (0)     tota
l: 87.4ms       remaining: 7m 16s
200:    test: 0.8429056 test1: 0.8130717        best: 0.8130717 (200)   tota
l: 12.4s        remaining: 4m 56s
400:    test: 0.8512820 test1: 0.8168956        best: 0.8169654 (396)   tota
l: 24.3s        remaining: 4m 38s
600:    test: 0.8580685 test1: 0.8188417        best: 0.8188417 (600)   tota
l: 36.6s        remaining: 4m 28s
800:    test: 0.8641671 test1: 0.8200149        best: 0.8200382 (792)   tota
l: 48.7s        remaining: 4m 15s
1000:   test: 0.8697181 test1: 0.8210826        best: 0.8211261 (993)   tota
l: 1m 1s        remaining: 4m 3s
1200:   test: 0.8753295 test1: 0.8217794        best: 0.8218322 (1188)  tota
l: 1m 13s       remaining: 3m 53s
1400:   test: 0.8809316 test1: 0.8226919        best: 0.8227121 (1399)  tota
l: 1m 27s       remaining: 3m 45s
1600:   test: 0.8863681 test1: 0.8231203        best: 0.8231777 (1543)  tota
l: 1m 41s       remaining: 3m 35s
Stopped by overfitting detector  (100 iterations wait)
```

In [ ]:
```
print(np.mean(scores40))
print(np.mean(scores41))
```

```
0.8096611422742035
0.6828921839980325
```

In [ ]:
```
pred4 = cat4.predict_proba(test)[:,1]
```

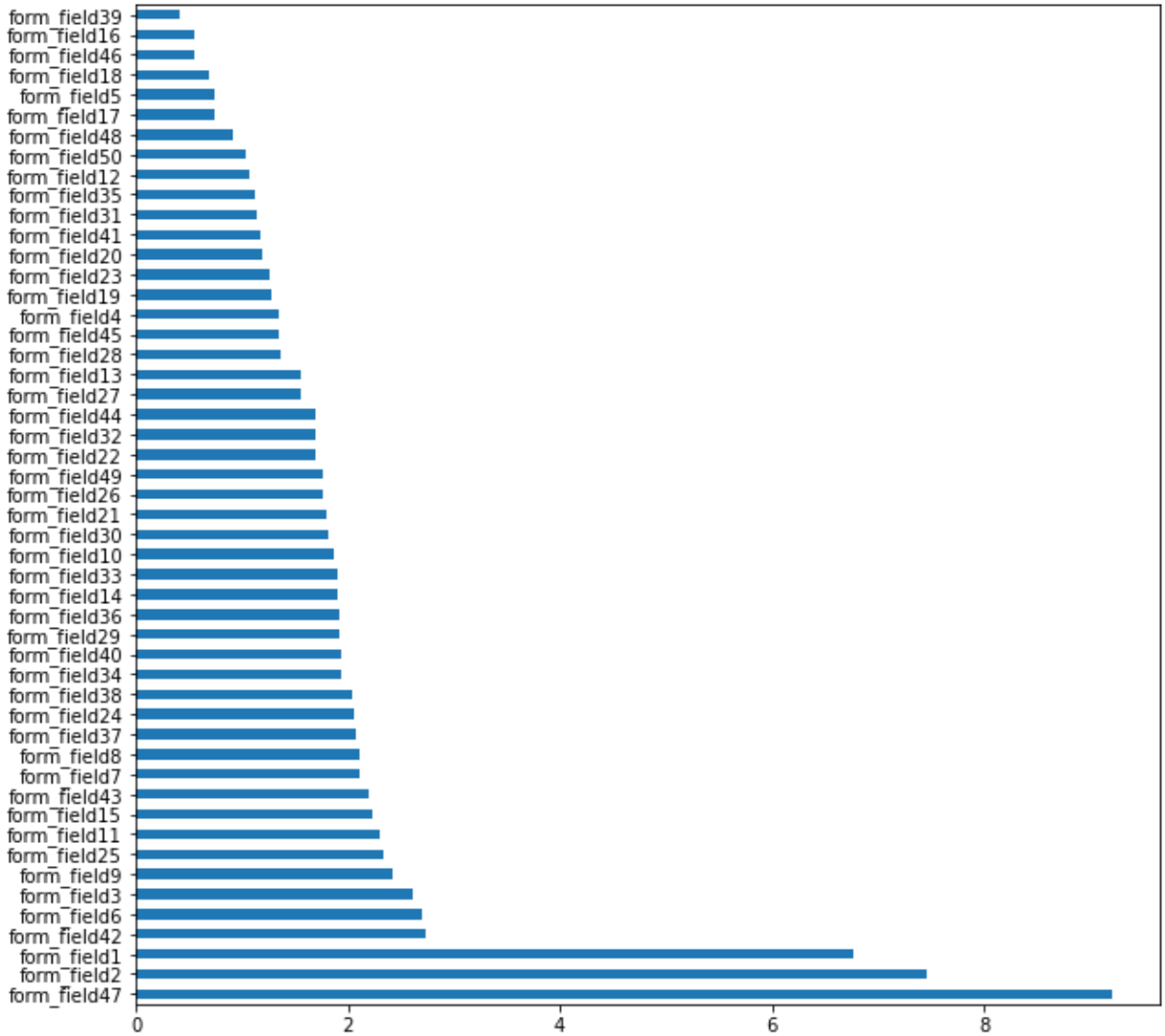
In [ ]:
```
print(cat4.feature_importances_)
```

```
[6.77136385 7.46620841 2.61815811 1.34200953 0.73694863 2.69707917
 2.11824181 2.10233648 2.42828359 1.86620978 2.29982856 1.06825823
 1.55724121 1.90030774 2.22595604 0.54425601 0.74708346 0.68797583
 1.27392072 1.19588189 1.8014224  1.69031352 1.25768361 2.05734229
 2.34242887 1.76860867 1.56197984 1.3692587  1.92297207 1.82258266
 1.13557115 1.69000183 1.8976538  1.93243687 1.11762714 1.91949144
 2.07746344 2.03871629 0.40825328 1.93047426 1.17280155 2.73952482
 2.19505261 1.68765745 1.35205541 0.55699034 9.20095781 0.91409259
 1.76036672 1.03066951]
```

In [ ]:
```
feat_imp = pd.Series(cat4.feature_importances_, index = X.columns)
```

In [ ]:
```
plt.figure(figsize = (10,10))
feat_imp.nlargest(50).plot(kind= 'barh')
```

Out[23]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f6b5e0bf9e8>
```




In [ ]:
```
X.drop('form_field39', axis = 1, inplace = True)
test.drop('form_field39', axis = 1, inplace = True)
```

In [ ]:
```
scores10 = []
scores11 = []
for fold,(tr_in,te_in) in enumerate(kf.split(X, y)):
    print(f"================================Fold{fold}============================
    X_train,X_test = X.iloc[tr_in],X.iloc[te_in]
    y_train,y_test = y.iloc[tr_in],y.iloc[te_in]
    cat4.fit(X_train,y_train,eval_set=[(X_train,y_train),(X_test,y_test)],early_s
    y_pred = cat4.predict(X_test)
    scores10.append(accuracy_score(y_test,y_pred))
    scores11.append(roc_auc_score(y_test,y_pred))
```

```
================================Fold0====================================
0:      test: 0.8072782 test1: 0.7821756        best: 0.7821756 (0)     tota
l: 62.6ms       remaining: 5m 12s
200:    test: 0.8429309 test1: 0.8126232        best: 0.8126480 (197)   tota
l: 12.7s        remaining: 5m 3s
400:    test: 0.8512761 test1: 0.8160451        best: 0.8160560 (396)   tota
l: 24.8s        remaining: 4m 43s
600:    test: 0.8583740 test1: 0.8181868        best: 0.8182023 (599)   tota
l: 37.2s        remaining: 4m 32s
800:    test: 0.8642650 test1: 0.8197542        best: 0.8197666 (799)   tota
l: 49.7s        remaining: 4m 20s
1000:   test: 0.8699764 test1: 0.8207474        best: 0.8207691 (990)   tota
l: 1m 2s        remaining: 4m 8s
1200:   test: 0.8754421 test1: 0.8215684        best: 0.8215854 (1199)  tota
l: 1m 14s       remaining: 3m 56s
1400:   test: 0.8812183 test1: 0.8221410        best: 0.8222465 (1361)  tota
l: 1m 27s       remaining: 3m 43s
1600:   test: 0.8864300 test1: 0.8228611        best: 0.8229216 (1588)  tota
l: 1m 39s       remaining: 3m 31s
```

In [ ]:
```
print(np.mean(scores10))
print(np.mean(scores11))
```

```
0.8097681703217519
0.6843677444985984
```

In [ ]:
```
pred1 = cat4.predict_proba(test)[:,1]
```

In [ ]:
```
lgbm = LGBMClassifier(boosting_type='gbdt', objective='binary', num_leaves=50,
                                  learning_rate=0.01, n_estimators=2000, max_depth=
```

In [ ]:
```
scores20 = []
scores21 = []
for fold,(tr_in,te_in) in enumerate(kf.split(X, y)):
    print(f"================================Fold{fold}============================
    X_train,X_test = X.iloc[tr_in],X.iloc[te_in]
    y_train,y_test = y.iloc[tr_in],y.iloc[te_in]
    lgbm.fit(X_train,y_train,eval_set=[(X_train,y_train),(X_test,y_test)],early_s
    y_pred = lgbm.predict(X_test)
    scores20.append(accuracy_score(y_test,y_pred))
    scores21.append(roc_auc_score(y_test,y_pred))
```

```
Early stopping, best iteration is:
[1094]  training's binary_logloss: 0.336182     valid_1's binary_logloss: 0.4
04563

================================Fold3====================================
Training until validation scores don't improve for 100 rounds.
[200]   training's binary_logloss: 0.406455     valid_1's binary_logloss: 0.4
12874
[400]   training's binary_logloss: 0.380678     valid_1's binary_logloss: 0.3
97644
[600]   training's binary_logloss: 0.365        valid_1's binary_logloss: 0.3
9467
[800]   training's binary_logloss: 0.352323     valid_1's binary_logloss: 0.3
93908
Early stopping, best iteration is:
[801]   training's binary_logloss: 0.352269     valid_1's binary_logloss: 0.3
93904
================================Fold4====================================
Training until validation scores don't improve for 100 rounds.
[200]   training's binary_logloss: 0.405855     valid_1's binary_logloss: 0.4
```

In [ ]:
```
print(np.mean(scores20))
print(np.mean(scores21))
```

```
0.8081785714285715
0.6853363349877205
```

In [ ]:
```
pred2 = lgbm.predict_proba(test)[:,1]
```

In [ ]:
```
submit['2'] = pred2
submit['4'] = pred4
submit['1'] = pred1
```

In [ ]:
```
submit['default_status'] = ((submit['2']* 0.2)+ ((submit['4'] * 0.7)+ submit['1']
```

In [ ]:
```
submit.drop(['2', '4', '1'], axis =1, inplace = True)
```

In [ ]:
```
submit.to_csv('submit2d5c.csv', index = False)
```