Account_scoring_plots

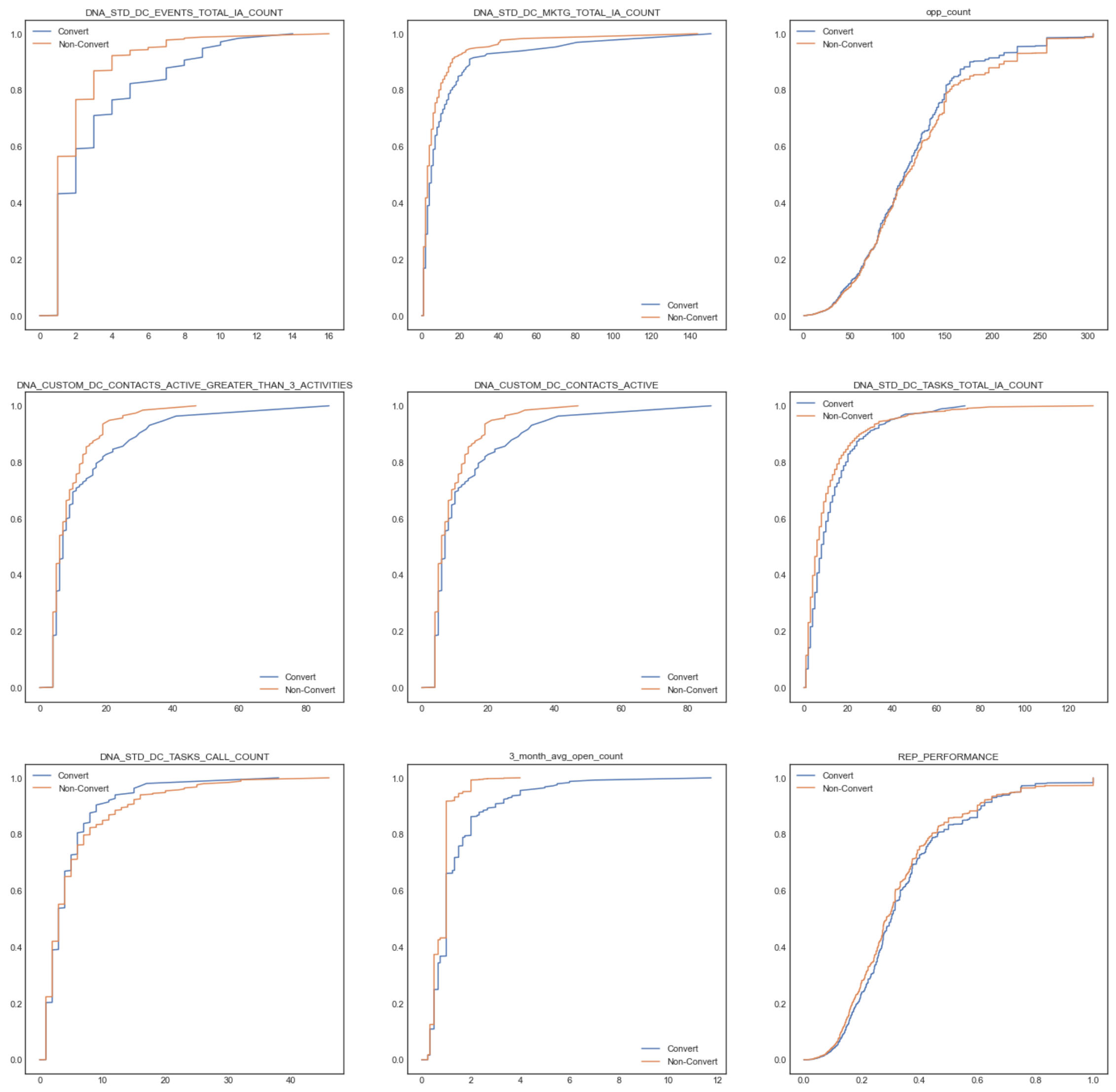
October 23, 2021

[1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.ticker import FixedLocator, FixedFormatter
from pandas_profiling import ProfileReport
import random
import seaborn as sns
import ptitprince as pt

cmap = sns.diverging_palette(220, 10, as_cmap=True)
sns.set(style = "darkgrid")
pd.options.mode.chained_assignment = None  # default='warn'
pd.set_option('display.max_columns', None)
plt.style.use('seaborn-whitegrid')
plt.style.use("seaborn-ticks")
plt.rcParams["xtick.direction"] = "in"
plt.rcParams["ytick.direction"] = "in"
plt.rcParams["font.size"] = 11.0
```

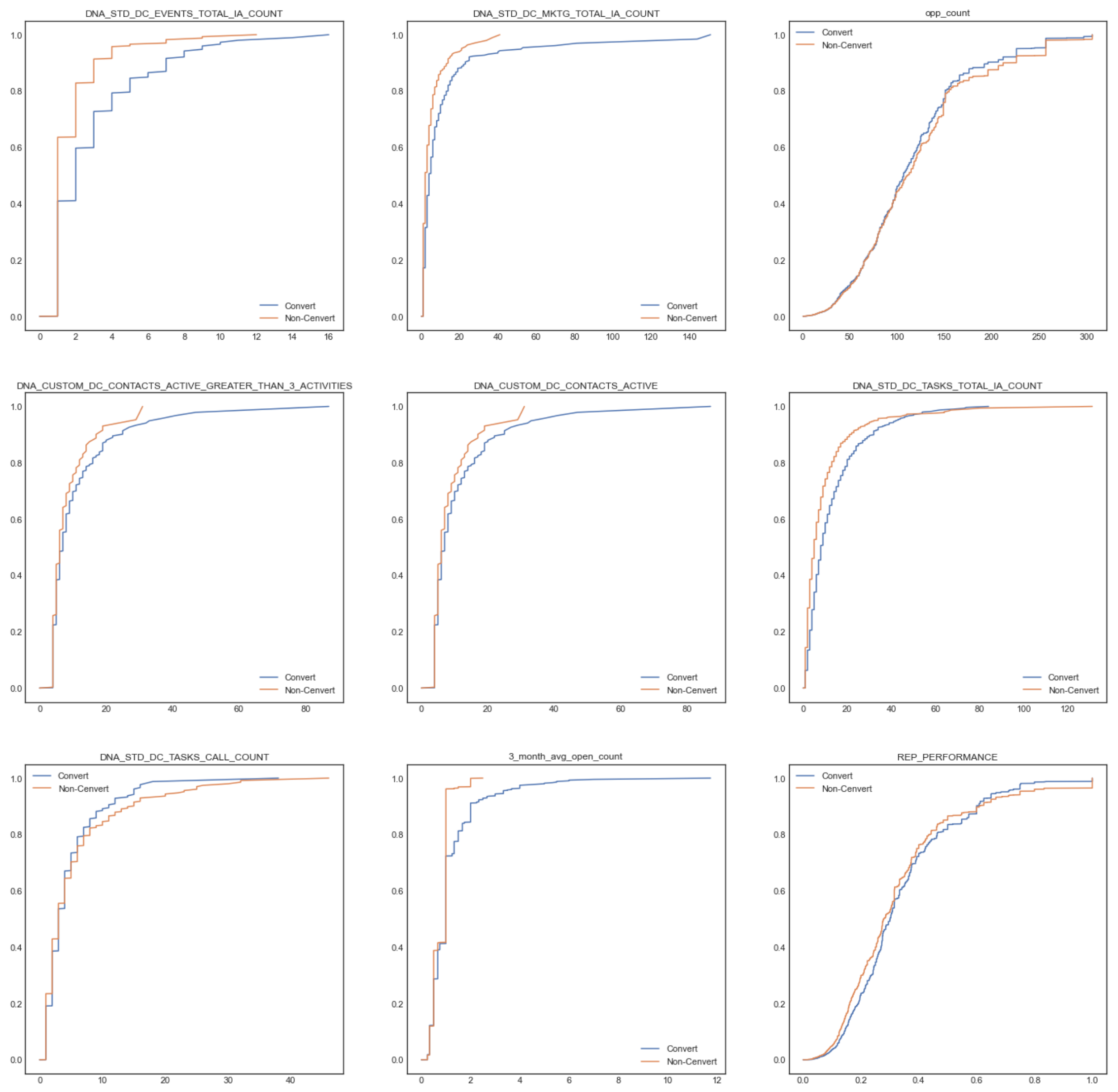
[2]:
```
df = pd.read_csv('../../data/Test_data_with_predictions_and_actual.csv')
```

[3]:
```
features = ['DNA_STD_DC_EVENTS_TOTAL_IA_COUNT',
         'DNA_STD_DC_MKTG_TOTAL_IA_COUNT',
         'opp_count',
         'DNA_CUSTOM_DC_CONTACTS_ACTIVE_GREATER_THAN_3_ACTIVITIES',
         'DNA_CUSTOM_DC_CONTACTS_ACTIVE',
         'DNA_STD_DC_TASKS_TOTAL_IA_COUNT',
         'DNA_STD_DC_TASKS_CALL_COUNT',
         '3_month_avg_open_count', 'REP_PERFORMANCE']
cat_fearure = ['DNA_CUSTOM_AC_ACCOUNT_TIER','DNA_STD_AC_INDUSTRY_GROUPS',]
true_label = ['Y']
predicted = ['Prediction_Class', 'Prediction_Score']
df = df[['ACCOUNT_ID'] + features + cat_fearure + predicted + true_label]
```

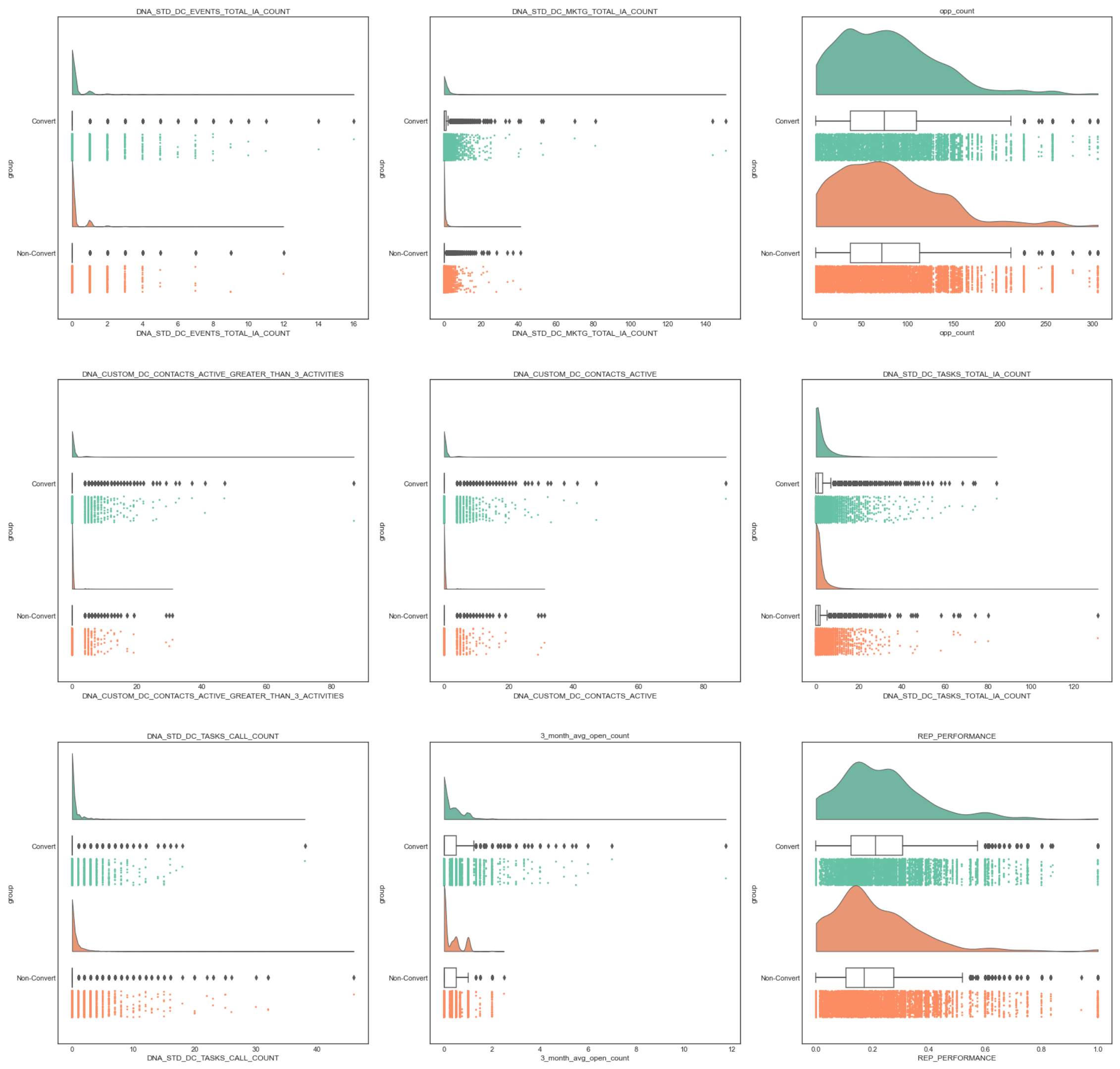
[4]:
```
plt.figure(figsize = (24,24))
for fi, feature in enumerate(features):
    plt.subplot(3,3,fi+1)
    for flag in [True, False]:
        x_ax = np.sort(df[df['Y']==flag][feature])
        y_ax = np.cumsum(x_ax)
        y_ax = y_ax/np.max(y_ax)
        plt.plot(x_ax,y_ax)
    plt.title(f'{feature}')
    plt.legend(['Convert', 'Non-Convert'])

plt.show()
```



[5]:
```
plt.figure(figsize = (24,24))
for fi, feature in enumerate(features):
    plt.subplot(3,3,fi+1)
    for flag in [True, False]:
        x_ax = np.sort(df[df['Prediction_Class']==flag][feature])
        y_ax = np.cumsum(x_ax)
        y_ax = y_ax/np.max(y_ax)
        plt.plot(x_ax,y_ax)
    plt.title(f'{feature}')
    plt.legend(['Convert', 'Non-Cenvert'])

plt.show()
```

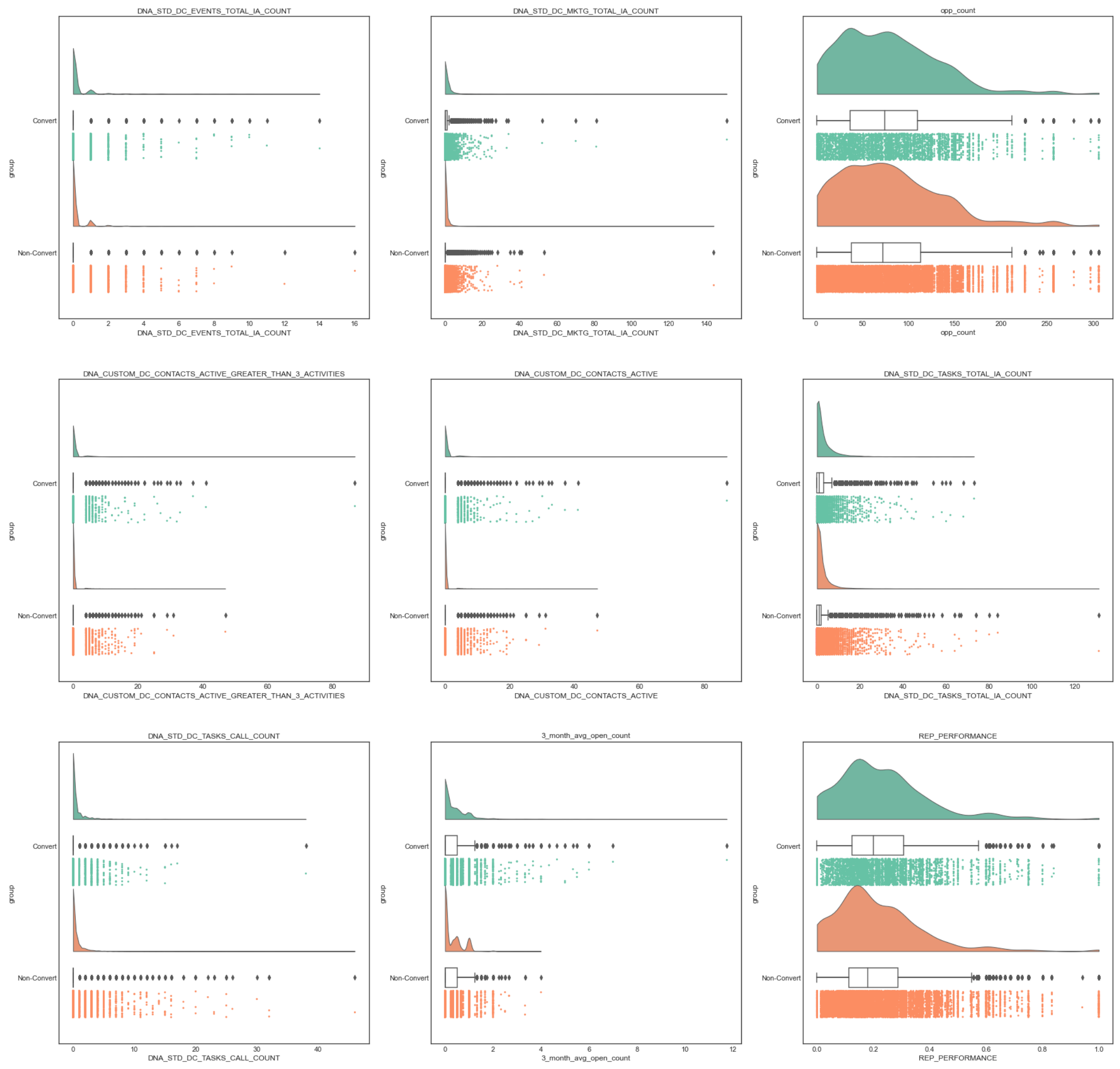


[6]:
```
fig, axes = plt.subplots(ncols=3, nrows=3, figsize=(30,30))
for fi, feature in enumerate(features):
    ax = axes.flat[fi]
    x = df[df['Prediction_Class']==True][feature].values.tolist()
    y = df[df['Prediction_Class']==False][feature].values.tolist()

    group = []
    score = []
    for i in range(0,len(x)):
        group.append('Convert')
        score.append(x[i])
    for i in range(0,len(y)):
        group.append('Non-Convert')
        score.append(y[i])
    disp_df = pd.DataFrame({"group":group,feature:score})

    dx = "group"; dy = feature; ort = "h"; pal = "Set2"; sigma = .2
    pt.RainCloud(x = dx, y = dy, data = disp_df, palette = pal, bw = sigma,␣
 ↪width_viol = 1.0, orient = ort, move = .2, ax=ax)
    ax.set_title(f"{feature}")
plt.show()
```



```
[7]: fig, axes = plt.subplots(ncols=3, nrows=3, figsize=(30,30))
     for fi, feature in enumerate(features):
         ax = axes.flat[fi]

         x = df[df['Y']==True][feature].values.tolist()
         y = df[df['Y']==False][feature].values.tolist()

         group = []
         score = []
         for i in range(0,len(x)):
             group.append('Convert')
             score.append(x[i])
         for i in range(0,len(y)):
             group.append('Non-Convert')
```

```
        score.append(y[i])
    disp_df = pd.DataFrame({"group":group,feature:score})

    dx = "group"; dy = feature; ort = "h"; pal = "Set2"; sigma = .2
    pt.RainCloud(x = dx, y = dy, data = disp_df, palette = pal, bw = sigma,␣
↪width_viol = 1.0, orient = ort, move = .2, ax=ax)
    ax.set_title(f"{feature}")

plt.show()
```



```
[ ]:
```