

Analysis_Out

July 22, 2022

An Exception was encountered at 'In [5]':

```
[1]: import warnings
warnings.filterwarnings('ignore')
```

```
[2]: import scrapbook as sb
import pandas as pd
import numpy as np
import seaborn as sns
import numpy as np
from statistics import mean , median
import matplotlib.pyplot as plt
```

1 Baseline

```
[3]: books = sb.read_notebooks("./BaseLine_Model_Output")
baseLine_data = []
for nb in books.notebooks:
    nbList=[nb.scrap['Stats Model MAE'].data,nb.scrap['Catboost MAE'].data,nb.
    ↳scrap['TabNet MAE'].data]
    baseLine_data.append(nbList)
df = pd.DataFrame(baseLine_data, columns = ["Stats Model","Catboost", "TabNet"])
baseline_data = np.array(baseLine_data)
stats = median(baseline_data[:,0])
catboost = median(baseline_data[:,1])
tabnet = median(baseline_data[:,2])
print(stats)
print(catboost)
print(tabnet)
```

```
0.3652752317201842
0.21916706077468695
0.25156091957515214
```

2 GAN Model

```
[4]: book = sb.read_notebooks("./GAN_Output")
gan_data = []
gan_mse = []
for nb in book.notebooks:
    metrics = nb.scrapes['GAN_1 Metrics'].data
    for i in range(1000):
        gan_mse.append(metrics[0][i])
    nbList = [nb.scrapes['GAN Model MSE'].data,
              nb.scrapes['GAN Model MAE'].data,
              nb.scrapes['GAN Model Euclidean distance'].data,
              nb.scrapes['GAN Model Manhattan Distance'].data]
    gan_data.append(nbList)

df = pd.DataFrame(gan_data, columns = ['MSE', 'MAE', 'Euclidean_
↳Distance', 'Manhattan Distance'])
display(df.style)
print("MEDIAN:")
print(df.median(axis = 0))
gan_data = np.array(gan_data)
gan_median = median(gan_data[:,1])
print(gan_median)
```

<pandas.io.formats.style.Styler at 0x7ffa131dac70>

MEDIAN:

MSE	0.204323
MAE	0.287760
Euclidean Distance	4.552393
Manhattan Distance	29.351488

dtype: float64
0.28775968519183714

3 GAN Model with TabNet generator

Execution using papermill encountered an exception here and stopped:

```
[5]: book = sb.read_notebooks("./GAN_Output")
gan_data = []
gan_mse = []
for nb in book.notebooks:
    metrics = nb.scrapes['GAN_2 Metrics'].data
    for i in range(1000):
        gan_mse.append(metrics[0][i])
    nbList = [nb.scrapes['GAN Model MSE'].data,
              nb.scrapes['GAN Model MAE'].data,
              nb.scrapes['GAN Model Euclidean distance'].data,
```

```

        nb.scrap['GAN Model Manhattan Distance'].data]
gan_data.append(nbList)

df = pd.DataFrame(gan_data, columns = ['MSE', 'MAE', 'Euclidean_
↳Distance', 'Manhattan Distance'])
display(df.style)
print("MEDIAN:")
print(df.median(axis = 0))
gan_data = np.array(gan_data)
gan_median = median(gan_data[:,1])
print(gan_median)

```

```

-----
KeyError                                Traceback (most recent call last)
Input In [5], in <cell line: 4>()
      3 gan_mse = []
      4 for nb in book.notebooks:
----> 5     metrics = nb.scrap['GAN_2 Metrics'].data
      6     for i in range(1000):
      7         gan_mse.append(metrics[0][i])

KeyError: 'GAN_2 Metrics'

```

4 ABC_GAN Analysis

4.1 ABC Pre-generator - Catboost

```

[ ]: book = sb.read_notebooks("./ABC_GAN_Catboost")
paramVal = [[1,1],[1,0.1],[1,0.01],[0.1,1],[0.1,0.1],[0.1,0.01],[0.01,1],[0.
↳01,0.1],[0.01,0.01]]
abc_mae = [[] for i in range(9)]
abc_mae_skip = [[] for i in range(9)]
abc_mae_mean = [[] for i in range(9)]
abc_mae_skip_mean = [[] for i in range(9)]
abc_weights = [[] for i in range(9)]
prior_model = [[] for i in range(9)]
abc_pre_generator = [[] for i in range(9)]

for nb in book.notebooks:
    metrics1 = np.array(nb.scrap['ABC_GAN_1 Metrics'].data)
    metrics3 = np.array(nb.scrap['ABC_GAN_3 Metrics'].data)
    paramVar = float(nb.papermill_dataframe.iloc[0]['value'])
    paramBias = float(nb.papermill_dataframe.iloc[1]['value'])
    #Divide data according to parameters
    for i in range(9):

```

```

if paramVar == paramVal[i][0] and paramBias == paramVal[i][1]:
    for j in range(100):
        abc_mae[i].append(metrics1[1,j])
        abc_mae_skip[i].append(metrics3[1,j])
        abc_weights[i].append(nb.scrap['Skip Connection Weight'].data)
        prior_model[i].append(nb.scrap['Prior Model MSE'].data)
        abc_pre_generator[i].append(nb.scrap['ABC Pre-generator MSE'].data)
        abc_mae_skip_mean[i].append(mean(metrics3[1,:]))
        abc_mae_mean[i].append(mean(metrics1[1,:]))

```

```

[ ]: data = [[] for i in range(9)]
for i in range(9):
    for j in range(len(abc_weights[i])):
        data[i].append([paramVal[i][0], paramVal[i][1],prior_model[i][j],
            abc_pre_generator[i][j],abc_weights[i][j],abc_mae_mean[i][j],abc_mae_skip_mean[i][j]])
        df = pd.DataFrame(data[i], columns = ['Variance','Bias','Prior Model MAE',
            'ABC pre-generator MAE','Skip Node
            weight','ABC GAN MAE','ABC_GAN MAE (skip connection)'])
        display(df.round(5))
        print(df.median(axis=0))
        print("-----")

```

```

[ ]: # Display Catboost Summary Tables
data = np.array(data)
catboostData = []
for i in range(9):
    catboostData.append([gan_median,catboost, paramVal[i][0],
        paramVal[i][1],median(data[i][:,3]),median(data[i][:,5]),median(data[i][:,6]),median(data[i][:,4])])
df = pd.DataFrame(catboostData, columns =
    ['GAN','catboost','Variance','Bias','Prior Model MAE','mGAN','skipGAN','Skip
    Node weight'])
display(df.round(5))

```

4.2 ABC Pre-generator - Stats

```

[ ]: book = sb.read_notebooks("./ABC_GAN_Stats")
paramVal = [[1,1],[0.1,1],[0.01,1],[1,0.1],[0.1,0.1],[0.01,0.1],[1,0.01],[0.1,0.01],[0.01,0.01]]
abc_mae = [[] for i in range(9)]
abc_mae_skip = [[] for i in range(9)]
abc_mae_mean = [[] for i in range(9)]
abc_mae_skip_mean = [[] for i in range(9)]
abc_weights = [[] for i in range(9)]
prior_model = [[] for i in range(9)]

```

```

abc_pre_generator = [[] for i in range(9)]

for nb in book.notebooks:
    metrics1 = np.array(nb.scrap['ABC_GAN_1 Metrics'].data)
    metrics3 = np.array(nb.scrap['ABC_GAN_3 Metrics'].data)
    paramVar = float(nb.papermill_dataframe.iloc[0]['value'])
    paramBias = float(nb.papermill_dataframe.iloc[1]['value'])
    #Divide data according to parameters
    for i in range(9):
        if paramVar == paramVal[i][0] and paramBias == paramVal[i][1]:
            for j in range(100):
                abc_mae[i].append(metrics1[1,j])
                abc_mae_skip[i].append(metrics3[1,j])
            abc_weights[i].append(nb.scrap['Skip Connection Weight'].data)
            prior_model[i].append(nb.scrap['Prior Model MSE'].data)
            abc_pre_generator[i].append(nb.scrap['ABC Pre-generator MSE'].data)
            abc_mae_skip_mean[i].append(mean(metrics3[1,:]))
            abc_mae_mean[i].append(mean(metrics1[1,:]))

```

```

[ ]: data = [[] for i in range(9)]
for i in range(9):
    for j in range(len(abc_weights[i])):
        data[i].append([paramVal[i][0], paramVal[i][1],prior_model[i][j],
            ↵
            ↪abc_pre_generator[i][j],abc_weights[i][j],abc_mae_mean[i][j],abc_mae_skip_mean[i][j]])

        df = pd.DataFrame(data[i], columns = ['Variance','Bias','Prior Model MAE',
            'ABC pre-generator MAE','Skip Node_
            ↪weight','ABC GAN MAE','ABC_GAN MAE (skip connection)'])
        display(df.round(5))
        print(df.median(axis=0))
        print("-----")

```

```

[ ]: # Display Stats Summary Tables
statsData = []
data = np.array(data)
for i in range(9):
    data[i] = np.array(data[i])
    statsData.append([gan_median,stats, paramVal[i][0],
        ↵
        ↪paramVal[i][1],median(data[i][:,3]),median(data[i][:,5]),median(data[i][:
        ↪,6]),median(data[i][:,4])])
    #statsData.append([gan_median,stats, paramVal[i][0], paramVal[i][1]])
df = pd.DataFrame(statsData, columns = ['GAN','stats','Variance','Bias','Prior_
    ↪Model MAE','mGAN','skipGAN','Skip Node weight'])
display(df.round(5))

```