

# Analysis\_Friedman1 (5 features + 100 datapoints)

January 19, 2022

```
[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
import matplotlib.pyplot as plt
```

## 1 Catboost Baseline

```
[3]: books = sb.read_notebooks("./BaseLine_Model_Output")
baseLine_data = []
for nb in books.notebooks:
    nbList=[nb.scrap['Catboost MSE'].data]
    baseLine_data.append(nbList)
df = pd.DataFrame(baseLine_data, columns = ["Catboost"])
baseLine_data = np.reshape(baseLine_data,(1,10))[0]
display(df)
mse = mean(baseLine_data)
print("Average MSE (Catboost Model): "+ str(mse))
```

```
Catboost
0  0.113988
1  0.179512
2  0.156973
3  0.143319
4  0.149362
5  0.154639
6  0.236295
7  0.140667
8  0.086507
9  0.296779
```

Average MSE (Catboost Model): 0.1658041049813956

## 2 GAN Analysis

```
[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("MEAN:")
print(df.mean(axis = 0))
gan_data = np.array(gan_data)
```

<pandas.io.formats.style.Styler at 0x7f8850e88940>

MEAN:

MSE	0.090416
MAE	0.197483
Euclidean Distance	1.297779
Manhattan Distance	3.949660

dtype: float64

## 3 ABC\_GAN Analysis

### 3.1 ABC Pre-generator

1. Prior Model is Catboost Model
2. ABC Pre-generator is Catboost Model with gaussian noise ->  $N(0, \text{variance})$  where variance : 1, 0.1, 0.01

```
[5]: book = sb.read_notebooks("./ABC_GAN_Catboost_Output")
paramVal = [1, 0.1, 0.01]
abc_mse = [[] for i in range(3)]
abc_mse_skip = [[] for i in range(3)]
abc_mse_mean = [[] for i in range(3)]
abc_mse_skip_mean = [[] for i in range(3)]
abc_weights = [[] for i in range(3)]
prior_model = [[] for i in range(3)]
abc_pre_generator = [[] for i in range(3)]
```

```

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'])

    #Divide data according to parameters
    for i in range(3):
        if paramVar == paramVal[i]:
            for j in range(100):
                abc_mse[i].append(metrics1[0,j])
                abc_mse_skip[i].append(metrics3[0,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_mse_mean[i].append(mean(metrics1[0,:]))
            abc_mse_skip_mean[i].append(mean(metrics3[0,:]))

```

```

[6]: for i in range(3):
    data = []
    for j in range(len(abc_weights[i])):
        data.append([paramVal[i],prior_model[i][j],
                    abc_pre_generator[i][j],abc_weights[i][j],
                    abc_mse_mean[i][j],abc_mse_skip_mean[i][j]])

    df = pd.DataFrame(data, columns = ['Variance','Prior Model MSE',
                                     'ABC pre-generator MSE','Skip Node_
→weight',
                                     'ABC_GAN MSE','ABC_GAN MSE (skip_
→connection)'])
    display(df.round(5))
    print(df.mean(axis=0))
    print("-----")

```

	Variance	Prior Model MSE	ABC pre-generator MSE	Skip Node weight \	
0	1	0.09503	1.05366	0.17071	
1	1	0.11280	1.01951	0.68027	
2	1	0.10933	1.29898	0.14053	
3	1	0.16532	1.80582	0.48354	
4	1	0.18313	0.77992	0.76103	
5	1	0.10814	1.05604	0.11168	
6	1	0.11635	1.31119	0.16299	
7	1	0.25473	1.08907	0.59638	
8	1	0.16082	0.76505	0.40229	
9	1	0.16134	1.13872	0.11723	
	ABC_GAN MSE	ABC_GAN MSE (skip connection)			
0	0.06444		0.05640		Pre-generator and GAN generator are comparable

1	0.05968	0.02703
2	0.05618	0.04994
3	0.27143	0.21673
4	0.27791	0.21161
5	0.01454	0.03663
6	0.04476	0.05351
7	0.21362	0.29436
8	0.05459	0.03621
9	0.09322	0.05120

Variance	1.000000
Prior Model MSE	0.146700
ABC pre-generator MSE	1.131796
Skip Node weight	0.362666
ABC_GAN MSE	0.115036
ABC_GAN MSE (skip connection)	0.103363
dtype: float64	

	Variance	Prior Model MSE	ABC pre-generator MSE	Skip Node weight \
0	0.1	0.08905	0.10427	0.55169
1	0.1	0.08641	0.09266	0.60263
2	0.1	0.13774	0.16070	0.14354
3	0.1	0.27338	0.29231	0.21075
4	0.1	0.17853	0.19730	0.09580
5	0.1	0.14023	0.15702	0.13731
6	0.1	0.06293	0.06070	0.13119
7	0.1	0.11731	0.10879	0.14659
8	0.1	0.20054	0.23918	0.20972
9	0.1	0.26067	0.23666	0.12495

	ABC_GAN MSE	ABC_GAN MSE (skip connection)
0	0.05159	0.04528
1	0.08046	0.07680
2	0.09973	0.07792
3	0.31136	0.16392
4	0.11026	0.08891
5	0.14030	0.12732
6	0.05697	0.04406
7	0.09193	0.11631
8	0.09566	0.09026
9	0.18944	0.20708

Variance	0.100000
Prior Model MSE	0.154679
ABC pre-generator MSE	0.164959
Skip Node weight	0.235419
ABC_GAN MSE	0.122770
ABC_GAN MSE (skip connection)	0.103786

dtype: float64

---

	Variance	Prior Model MSE	ABC pre-generator MSE	Skip Node weight \
0	0.01	0.12813	0.13049	0.00000
1	0.01	0.28523	0.28260	0.00000
2	0.01	0.23795	0.23835	0.00000
3	0.01	0.24982	0.25087	0.41817
4	0.01	0.40875	0.40620	0.00000
5	0.01	0.54401	0.54447	0.12631
6	0.01	0.23755	0.23860	0.40270
7	0.01	0.16374	0.16546	0.00000
8	0.01	0.14907	0.14708	0.01781
9	0.01	0.16936	0.16888	0.01309

	ABC_GAN MSE	ABC_GAN MSE (skip connection)
0	0.13111	0.12829
1	0.24372	0.28517
2	0.24358	0.23825
3	0.17479	0.20875
4	0.27700	0.40885
5	0.47027	0.53416
6	0.18846	0.21424
7	0.12366	0.16422
8	0.18739	0.14593
9	0.12798	0.17213

Variance	0.010000
Prior Model MSE	0.257362
ABC pre-generator MSE	0.257301
Skip Node weight	0.097808
ABC_GAN MSE	0.216798
ABC_GAN MSE (skip connection)	0.250000

dtype: float64

---