Classification Algorithms:

1. Logictic Regression
2. Naive Bayes
3. K-Nearest Neighbors
4. Support Vector Machines
5. Decision Tree
6. XGBoost
7. LightGBM
8. Random Forest Classifiers

In [3]:
```
import numpy  as np
import pandas as pd


from models.classifiers import Classifiers
from models.custom_ml_classifiers import CustomMLClassifiers

import warnings
warnings.filterwarnings('ignore')
```

In [4]:
```
# data paths
path_train_baseline = "./data/train_imputed_onehotencoded.csv"
```

In [5]:
```
# read data into data frame
df_train = pd.read_csv(path_train_baseline)
df_train.head()
```

Out[5]:

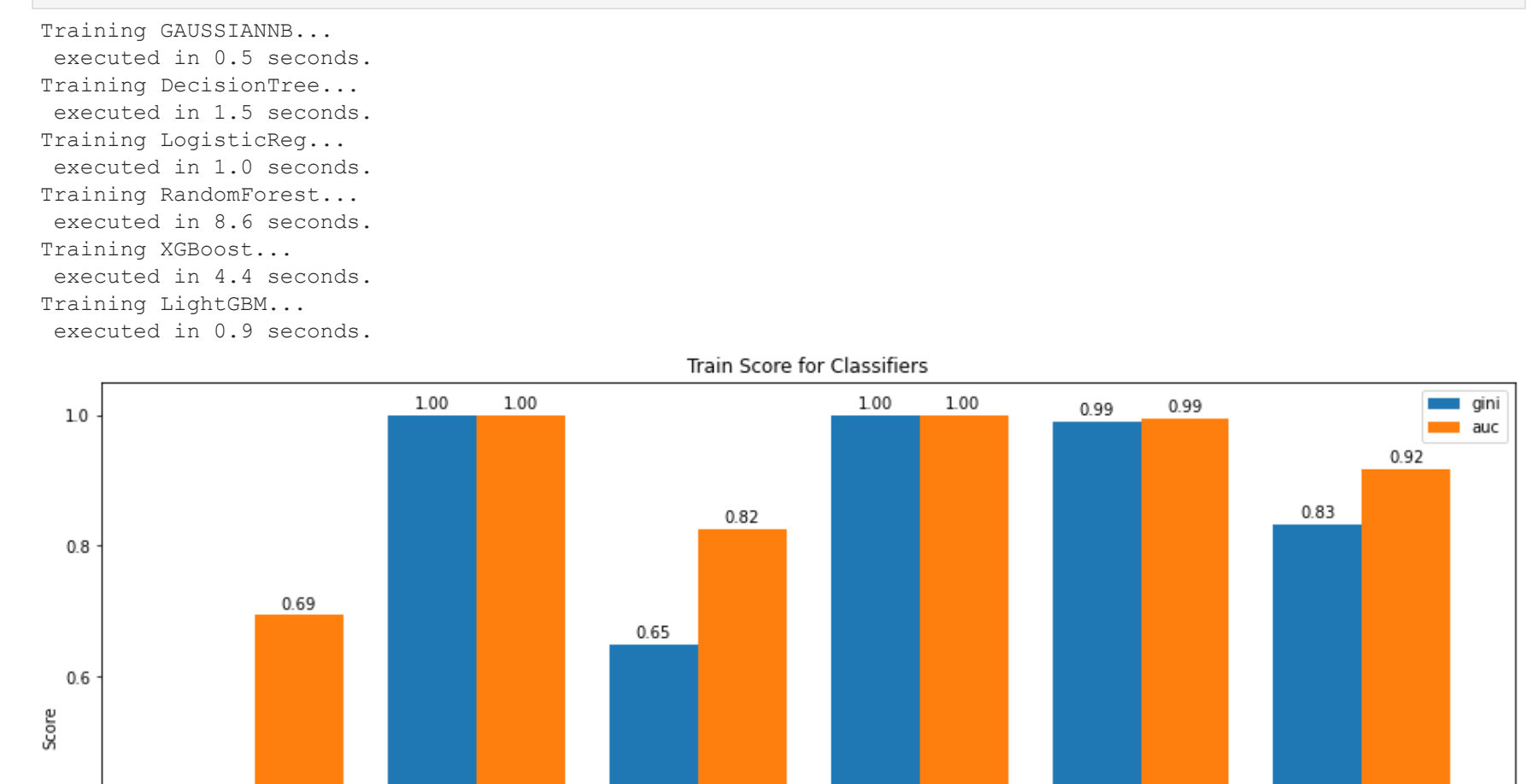
	ID	VAR0	VAR1	VAR2	VAR3	VAR4	VAR5	VAR6	VAR7	VAR8	...	CAT158	CAT159	CAT160	CA
0	0	0.799805	0.184174	0.000000	0.000000	0.168984	0.037037	0.000000	0.200195	0.788112	...	0.0	0.0	0.0	
1	1	0.725049	0.180594	0.000000	0.000000	0.165312	0.090790	0.054237	0.255107	0.722679	...	0.0	0.0	0.0	
2	2	0.799805	0.199255	0.000000	0.000000	0.167613	0.000000	0.000000	0.200195	0.788112	...	0.0	0.0	0.0	
3	3	0.677137	0.209870	0.166667	0.181818	0.190511	0.222222	0.125000	0.322863	0.788112	...	0.0	0.0	0.0	
4	4	0.725049	0.180594	0.000000	0.000000	0.165312	0.000000	0.054237	0.255107	0.722679	...	0.0	0.0	0.0	

5 rows × 206 columns

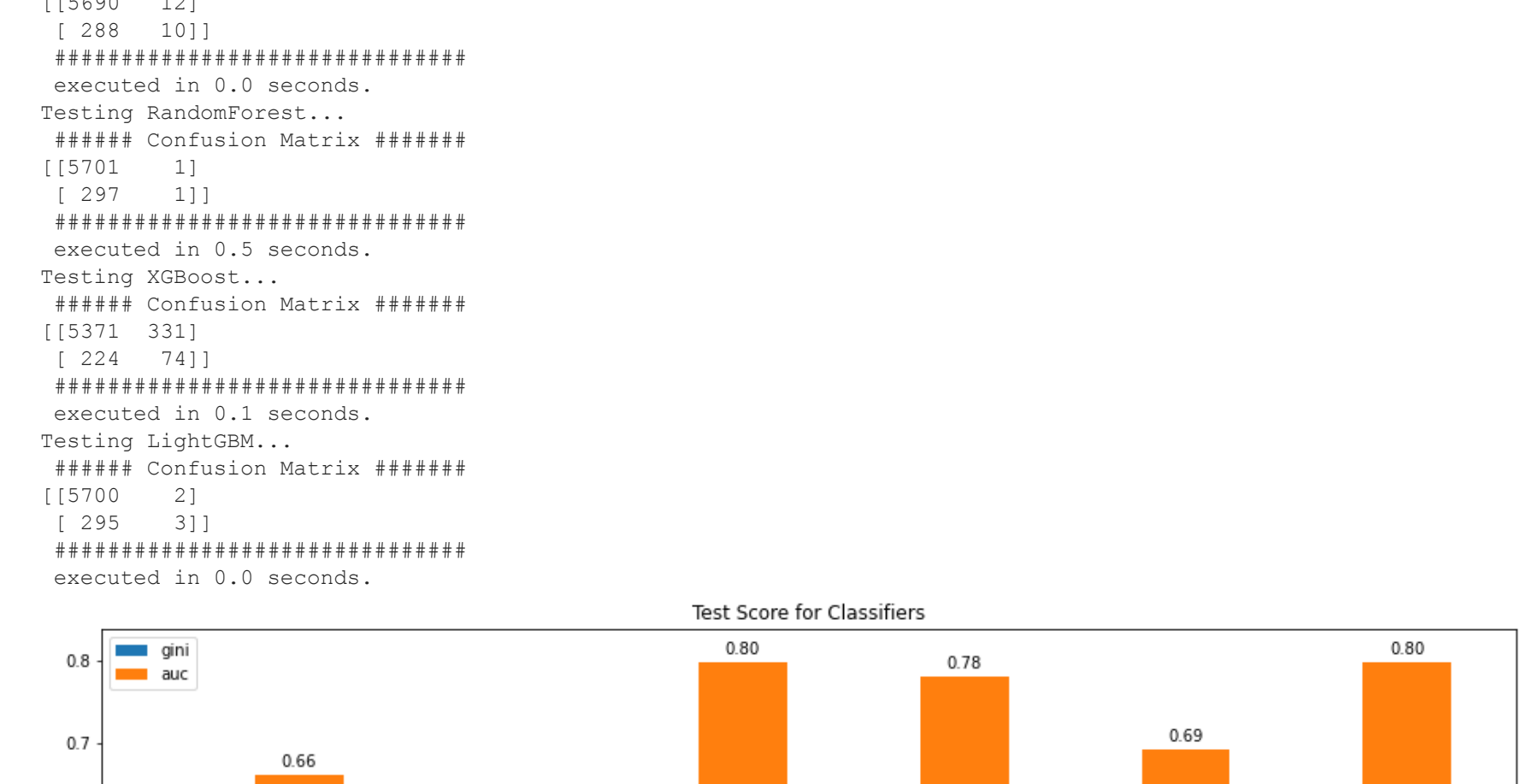
In [6]:
```
col     = "TARGET"
exclude = ["ID"]

# drop excluded columns and target column for X value
X = df_train.drop(exclude + [col], axis=1).values

# get target values
y = df_train[col].values

classifiers  = Classifiers(X, y, test_size=0.20, classifiers = CustomMLClassifiers.list(set_class_weight=True))
```

In [7]:
```
scores_train = classifiers.compare_train(cv=5)
score_test   = classifiers.compare_test()
```

```
Training GAUSSIANNB...
 executed in 0.5 seconds.
Training DecisionTree...
 executed in 1.5 seconds.
Training LogisticReg...
 executed in 1.0 seconds.
Training RandomForest...
 executed in 8.6 seconds.
Training XGBoost...
 executed in 4.4 seconds.
Training LightGBM...
 executed in 0.9 seconds.
```




```
Testing GAUSSIANNB...
 ###### Confusion Matrix #######
[[ 302 5400]
 [  13  295]]
################################
 executed in 0.1 seconds.
Testing DecisionTree...
 ###### Confusion Matrix #######
[[5419  283]
 [ 255   43]]
################################
 executed in 0.0 seconds.
Testing LogisticReg...
 ###### Confusion Matrix #######
[[5690   12]
 [ 288   10]]
################################
 executed in 0.0 seconds.
Testing RandomForest...
 ###### Confusion Matrix #######
[[5701    1]
 [ 297    1]]
################################
 executed in 0.5 seconds.
Testing XGBoost...
 ###### Confusion Matrix #######
[[5371  331]
 [ 224   74]]
################################
 executed in 0.1 seconds.
Testing LightGBM...
 ###### Confusion Matrix #######
[[5700    2]
 [ 295    3]]
################################
 executed in 0.0 seconds.
```



In [8]:
```
scores_train = classifiers.hyperparams_search(n_iter=50, cv=2)
score_test   = classifiers.compare_test()
```