POLITECNICO
MILANO 1863

Mobile Radio Networks Project

User QoE Estimation

Sajad Hamzenejadi
Ashkan Ramezani

Contact: sajad.hamzenejadi@mail.polimi.it

Outline

1. Datasets

Dataset Name	Dataset Shape	Decryption	NaN - Values
Dataset 1	(18970,12)	Original dataset for training	No
Dataset 2	(18970,11)	Generated using dataset 1 during feature engineering	No
Dataset 3	(18970,23)	Dataset 1 and 2 combined	No
Ground_Truth	(18970, 1)	Ground truth containing training targets Class 0 #: 12758 Class 1 #: 6212	No
Test_dataset 1	(4743,12)	Original dataset for testing	No
Test_dataset 2	(4743,11)	Generated using test_dataset 1 during feature engineering	No
Test_dataset 3	(4743,23)	Test_dataset 1 and 2 combined	No
Test_Ground_Truth	(4743,1)	Ground truth containing testing targets Class 0 #: 3221 Class 1 #: 1522	No

2. Algorithms

2.1. Decision Tree: A decision tree is a tree-like model where each internal node represents a feature and a decision based on its value, leading to different branches. At the leaf nodes, the outcome (classification label) is assigned based on the majority class of the training examples that reach that leaf. The tree is built recursively by selecting the best feature and split point at each node to maximize information gain (or minimize impurity) until a stopping criterion is met [1].

2.2. Random Forest: A random forest is an ensemble of decision trees. It builds multiple decision trees using bootstrapped subsets of the training data. Each tree is trained independently with a subset of features. During prediction, each tree "votes" for a class, and the class with the most votes becomes the final prediction. This reduces overfitting and increases predictive accuracy [2].

2.3. XGBoost (Extreme Gradient Boosting): XGBoost is an advanced boosting algorithm. It combines weak learners (simple models) into a strong model. It starts with an initial prediction (usually the mean of the target values) and then builds subsequent trees to correct the errors made by previous trees. It uses gradient descent to optimize the model's weights. Regularization terms are added to control complexity and prevent overfitting [3].

2.4. Grid Search: Grid search is a hyperparameter tuning technique where a predefined grid of hyperparameter values is specified. The algorithm systematically evaluates each combination of values using cross-validation. It searches through the entire grid to find the combination that yields the best performance according to a chosen evaluation metric [4].

2.5. Bayesian Optimization: Bayesian optimization is a smarter way of tuning hyperparameters. It models the unknown function (objective function) that maps hyperparameters to their performance using a probabilistic model (often Gaussian Process). The model estimates the function's behavior and uncertainty. The acquisition function (e.g., Expected Improvement) guides the search by choosing the next set of hyperparameters that is likely to improve the model's performance [5].

3. Feature Engineering: Generating Dataset 2

1. Avg_Youtube_Time_UMTS: This feature is calculated by dividing the cumulative YouTube session time for UMTS by 30 (days) to get the average daily YouTube DL time.

2. Avg_Youtube_Volume_UMTS: Similar to the previous feature, this is the average daily YouTube DL volume for UMTS.

3. Avg_Youtube_Time_LTE: Average daily YouTube DL time for LTE.

4. Avg_Youtube_Volume_LTE: Average daily YouTube DL volume for LTE.

5. Service_Availability_Ratio_UMTS: This ratio is calculated by dividing the cumulative full service time for UMTS by 30 (days) and then dividing it by the sum of cumulative limited service time and cumulative no service time for UMTS (divided by 30).

6. Service_Availability_Ratio_LTE: Similar to the previous feature, this is the service availability ratio for LTE.

7. Total_Service_Time_UMTS: This is the total service time for UMTS, calculated as the sum of cumulative full service time, cumulative limited service time, and cumulative no service time, all divided by 30 (days).

8. Total_Service_Time_LTE: Similar to the previous feature, this is the total service time for LTE.

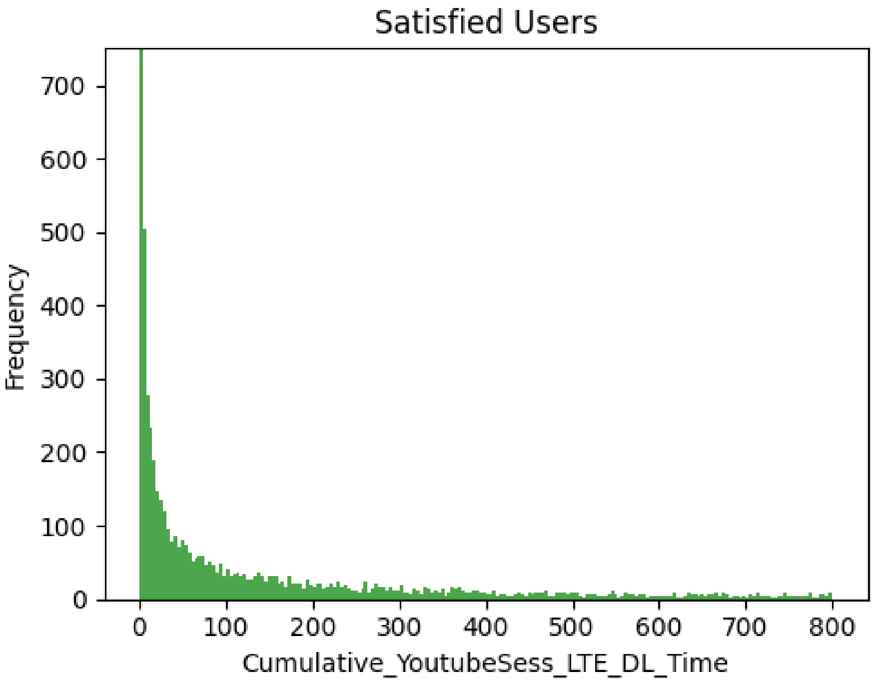
9. Youtube_Time_Percentage_UMTS: This feature calculates the percentage of time spent on YouTube compared to the total service time for UMTS.

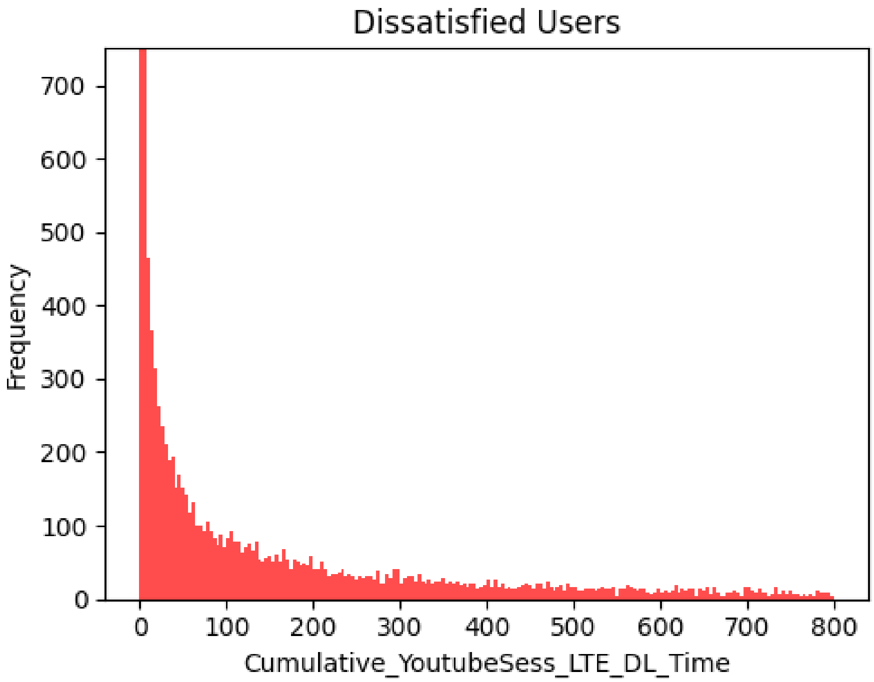
10. Youtube_Time_Percentage_LTE: Similar to the previous feature, this is the YouTube time percentage for LTE.

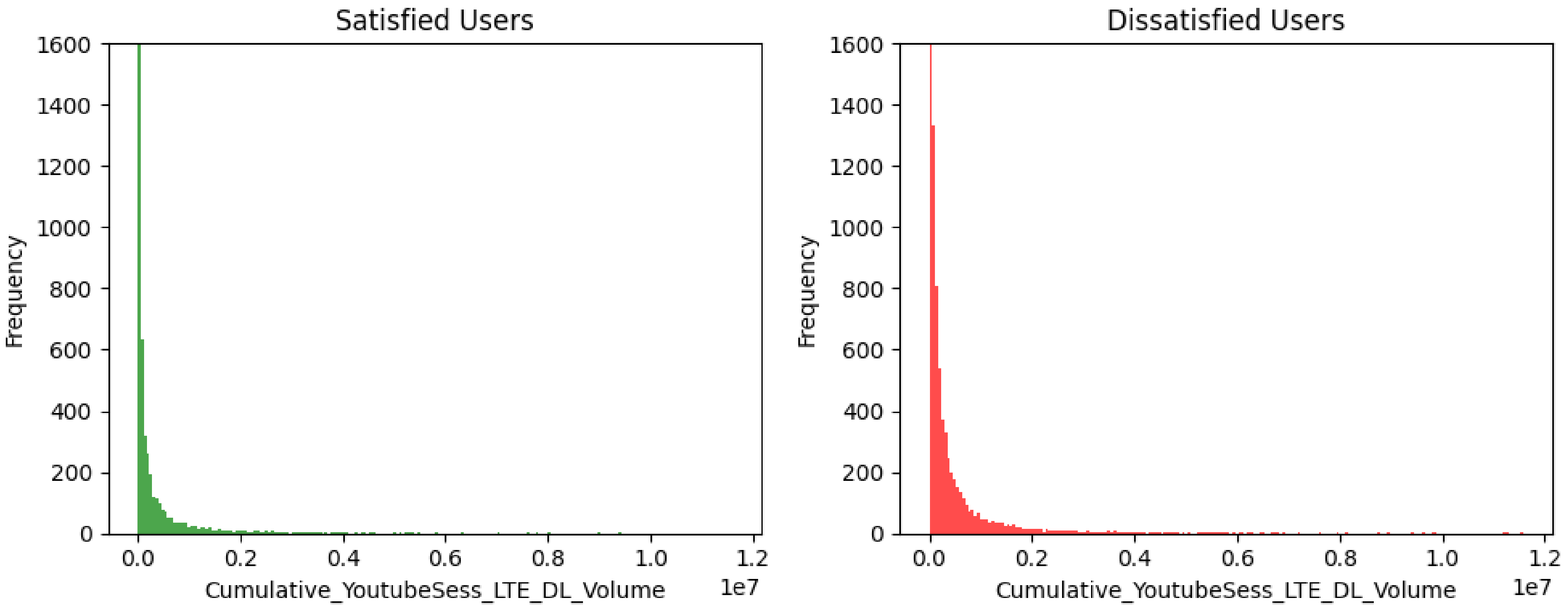
11. Preferred_Network_Type: This binary feature indicates whether LTE is the preferred network type based on which has a higher total service time (1 for LTE, 0 for UMTS).

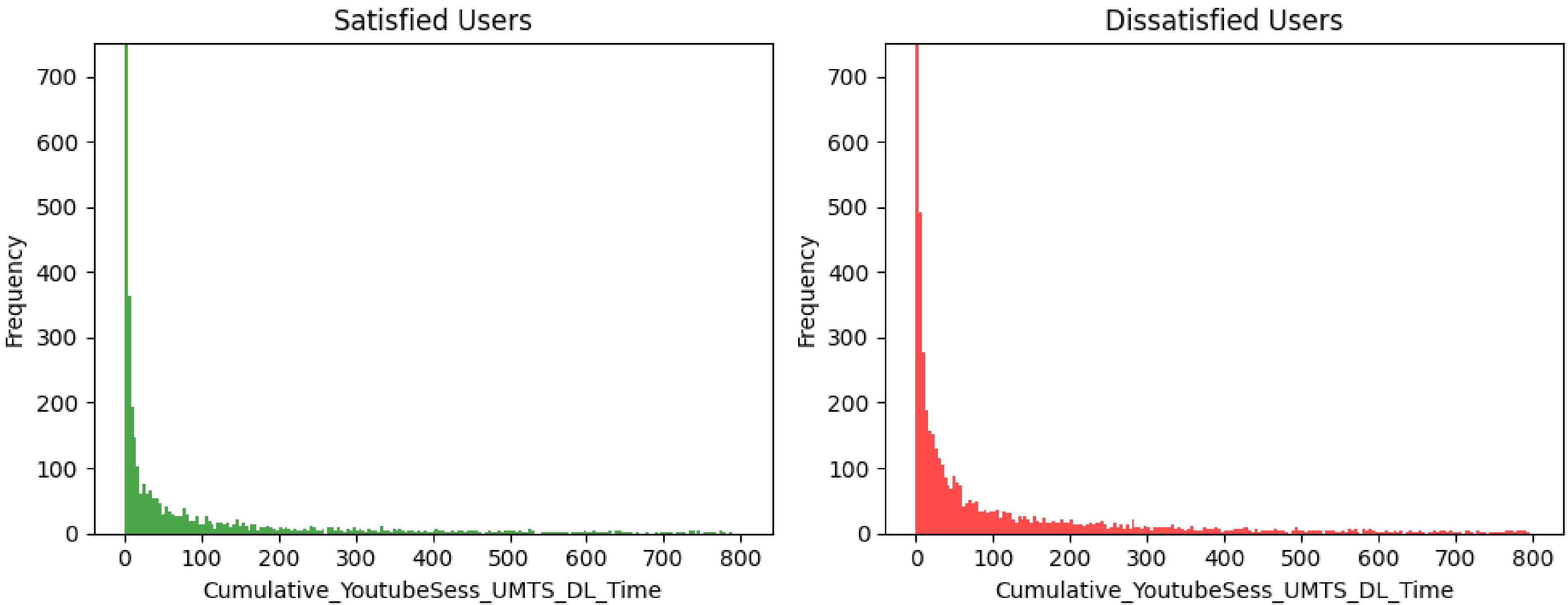
4. Feature Analysis

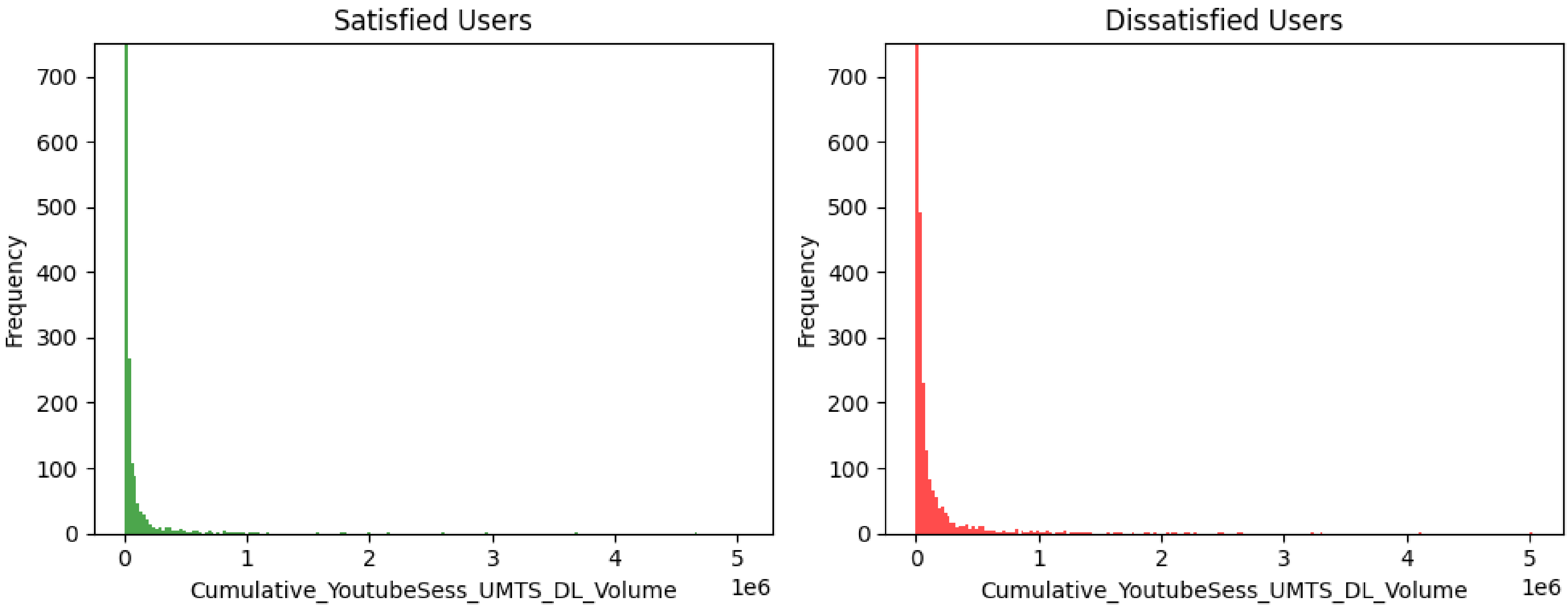
4.1. Relation between user QoE and data distribution: Histogram

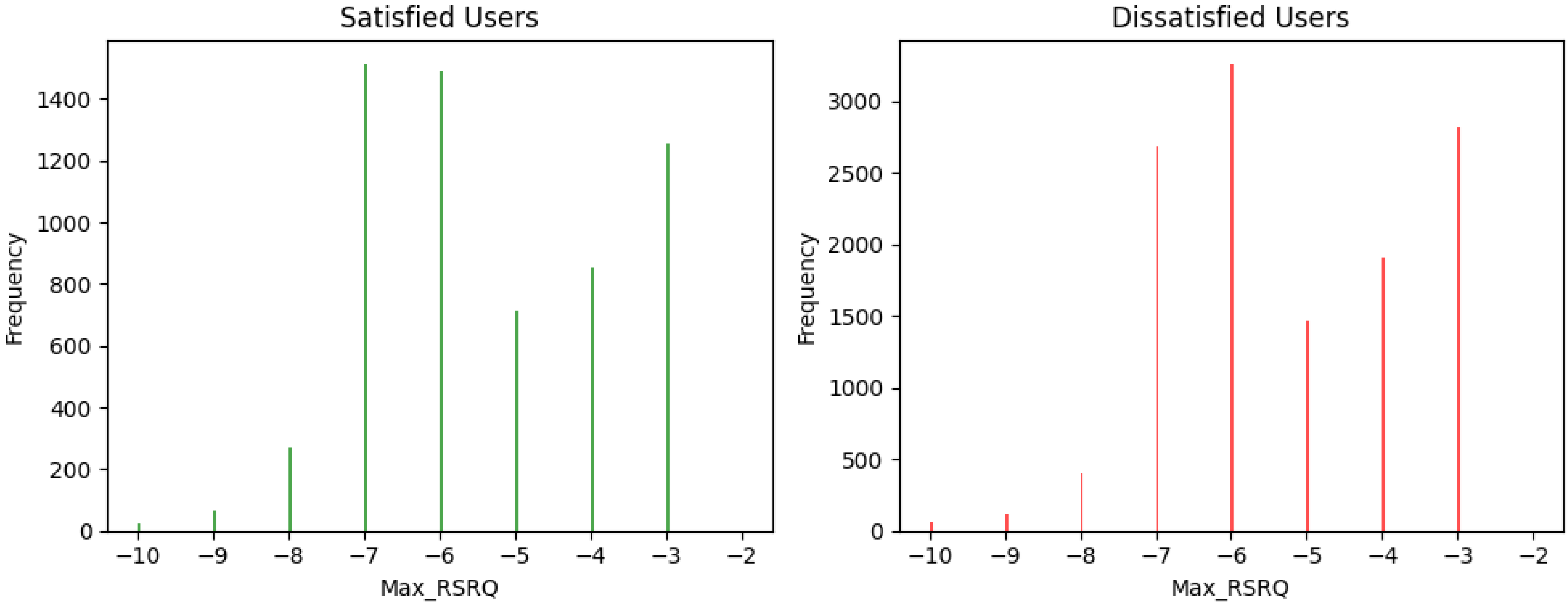




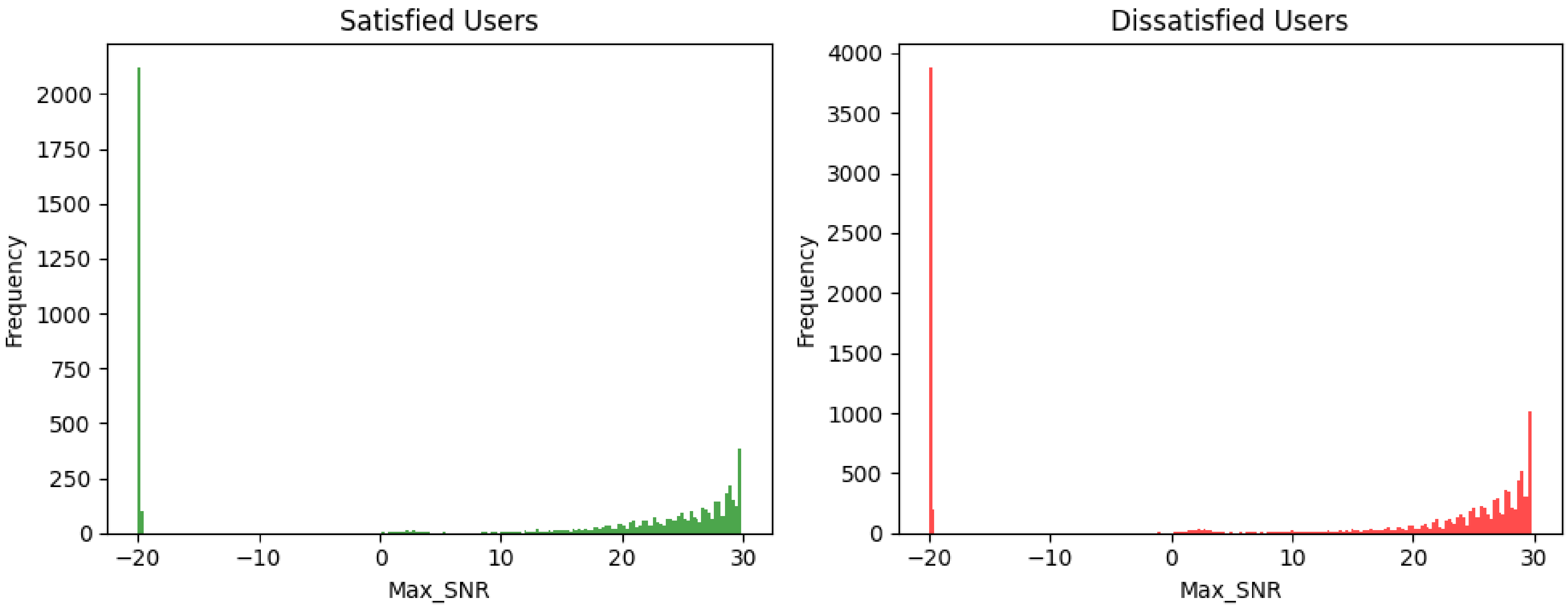

Satisfied Users
Frequency
1600
1400
1200
1000
800
600
400
200
0
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1e7
Cumulative_YoutubeSess_LTE_DL_Volume
Dissatisfied Users
Frequency
1600
1400
1200
1000
800
600
400
200
0
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1e7
Cumulative_YoutubeSess_LTE_DL_Volume

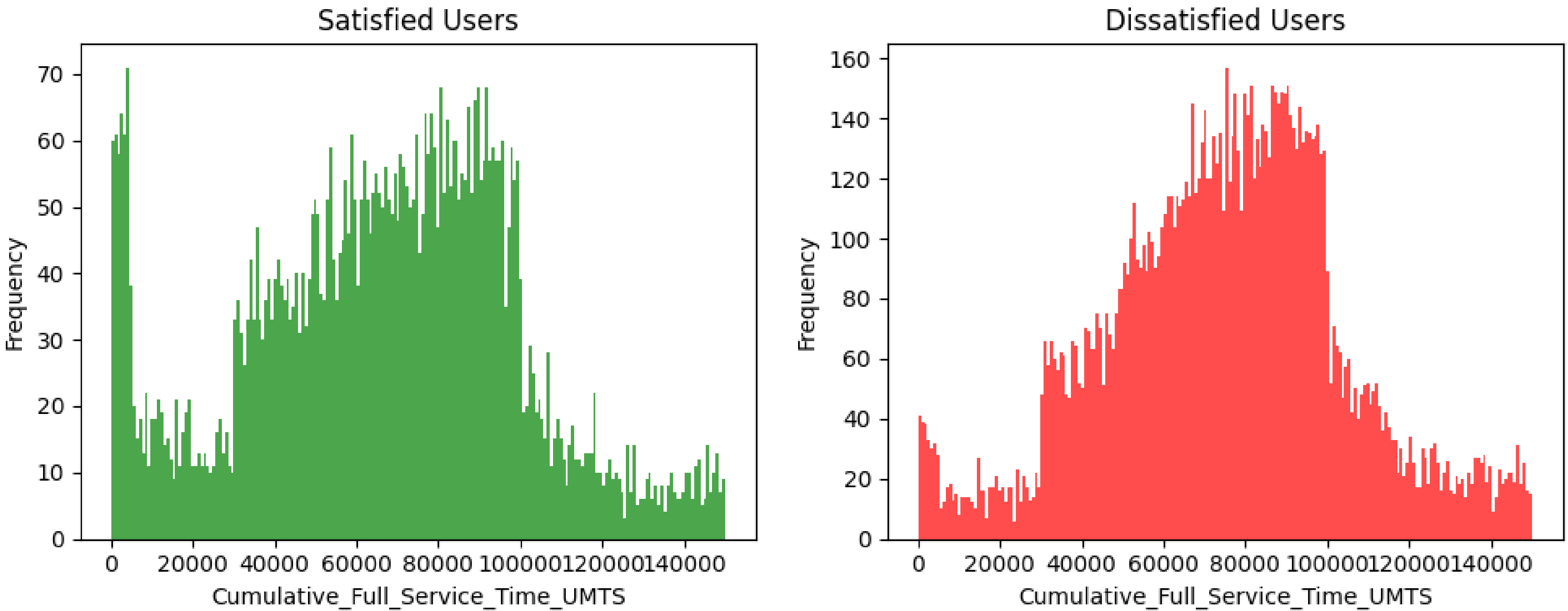

Satisfied Users
Dissatisfied Users
Frequency
Frequency
700
600
500
400
300
200
100
0
0
100
200
300
400
500
600
700
800
Cumulative_YoutubeSess_UMTS_DL_Time
Cumulative_YoutubeSess_UMTS_DL_Time


Satisfied Users
Dissatisfied Users
Frequency
0
100
200
300
400
500
600
700
0
1
2
3
4
5
1e6
Cumulative_YoutubeSess_UMTS_DL_Volume

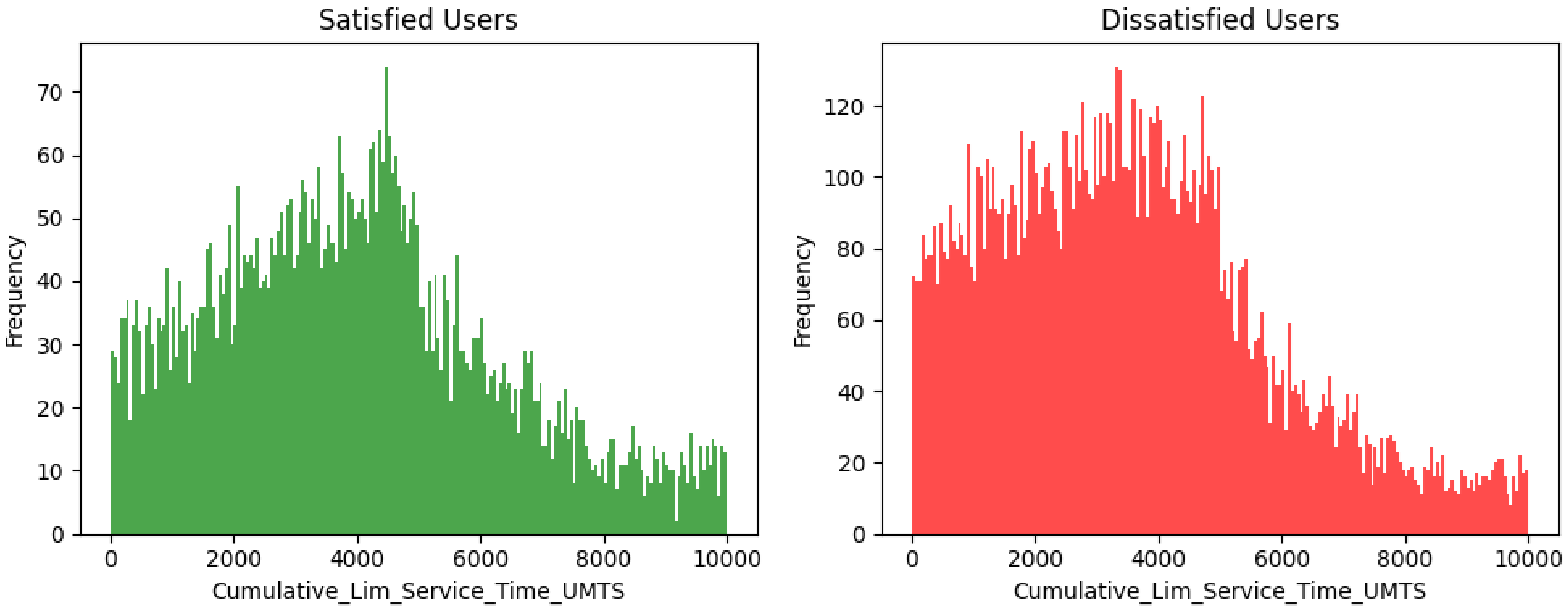


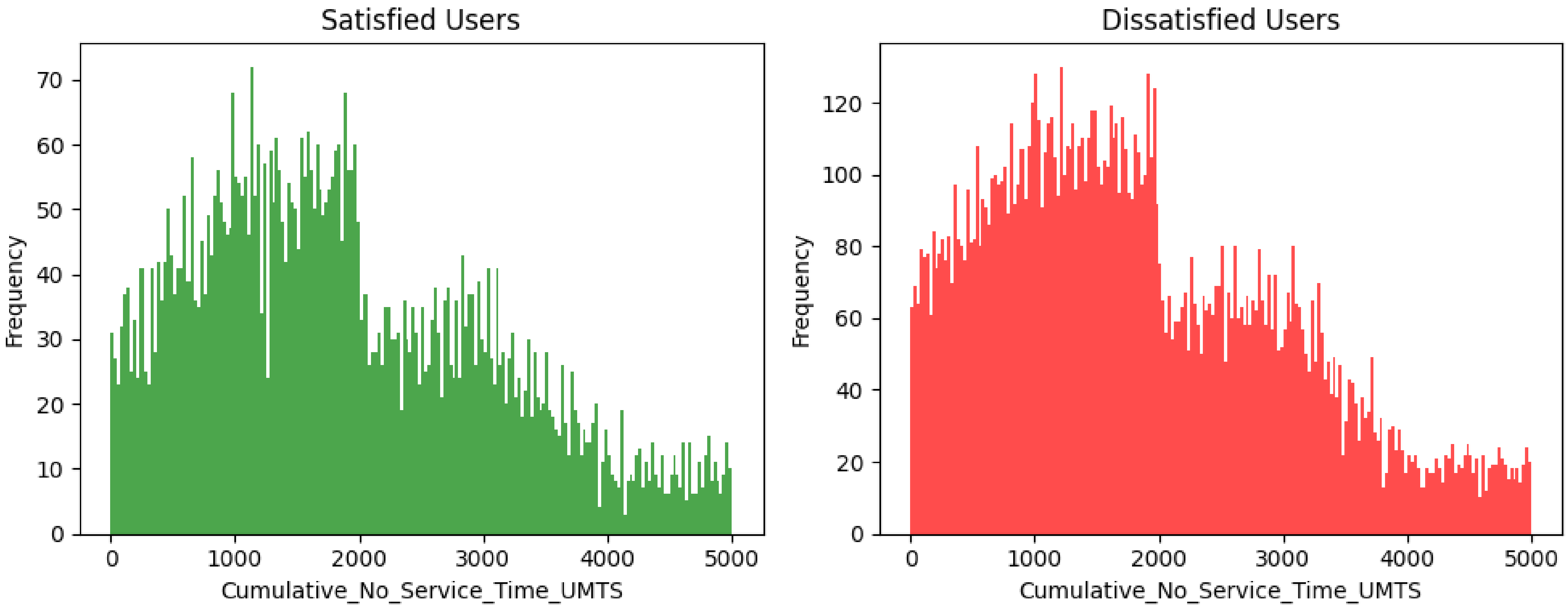
Satisfied Users
Frequency
Max_RSRQ
0
200
400
600
800
1000
1200
1400
−10
−9
−8
−7
−6
−5
−4
−3
−2
Dissatisfied Users
Frequency
Max_RSRQ
0
500
1000
1500
2000
2500
3000
−10
−9
−8
−7
−6
−5
−4
−3
−2

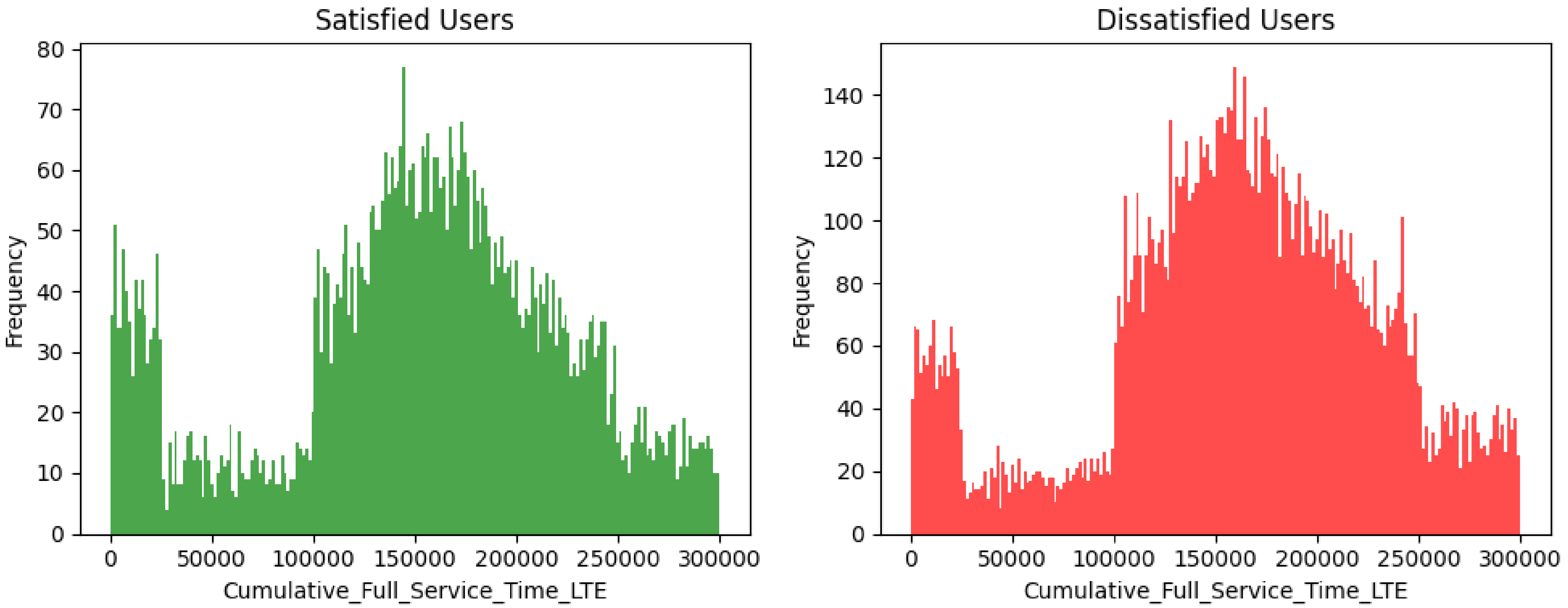

Satisfied Users
Frequency
Max_SNR
Dissatisfied Users
Frequency
Max_SNR



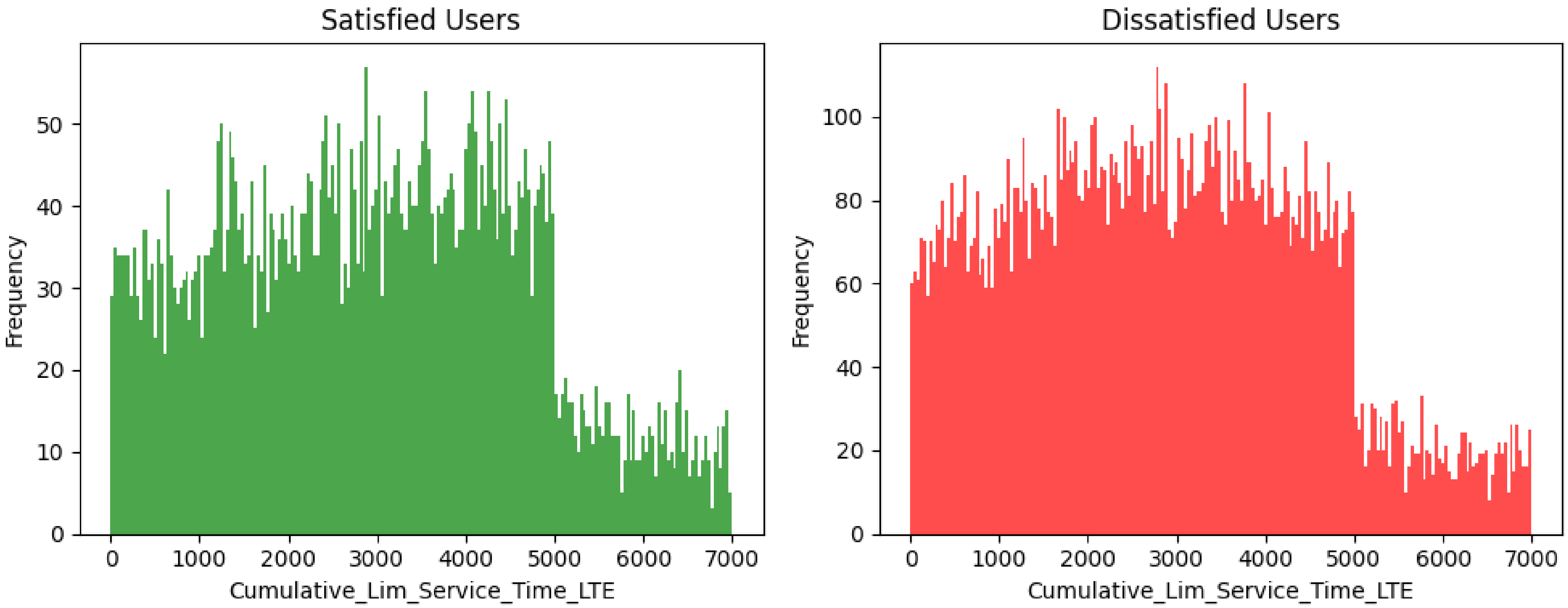
Satisfied Users
Frequency
Cumulative_Full_Service_Time_UMTS
0
10
20
30
40
50
60
70
0
20000
40000
60000
80000
100000
120000
140000
Dissatisfied Users
Frequency
Cumulative_Full_Service_Time_UMTS
0
20
40
60
80
100
120
140
160
0
20000
40000
60000
80000
100000
120000
140000

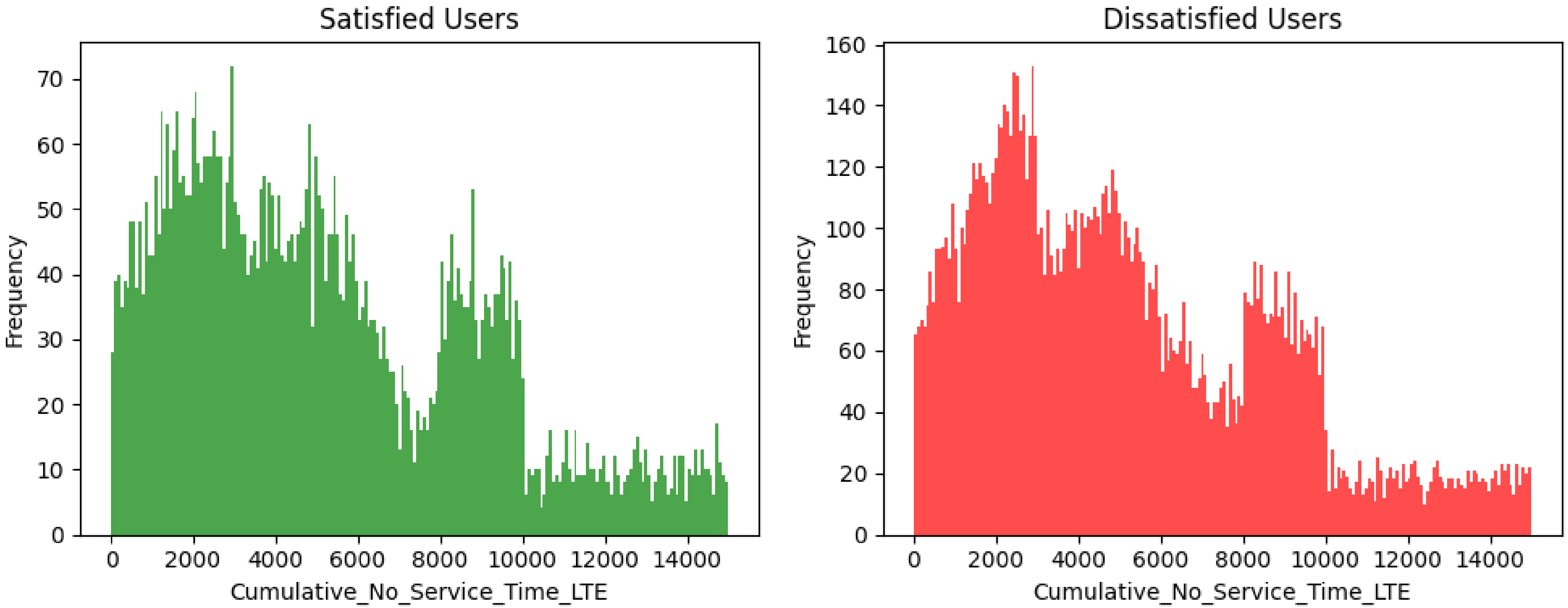

Satisfied Users
Frequency
Cumulative_Lim_Service_Time_UMTS
0
10
20
30
40
50
60
70
0
2000
4000
6000
8000
10000
Dissatisfied Users
Frequency
Cumulative_Lim_Service_Time_UMTS
0
20
40
60
80
100
120
0
2000
4000
6000
8000
10000


Satisfied Users
Frequency
Cumulative_No_Service_Time_UMTS
Dissatisfied Users
Frequency
Cumulative_No_Service_Time_UMTS



Satisfied Users
Frequency
Cumulative_Full_Service_Time_LTE
0
10
20
30
40
50
60
70
80
0
50000
100000
150000
200000
250000
300000
Dissatisfied Users
Frequency
Cumulative_Full_Service_Time_LTE
0
20
40
60
80
100
120
140
0
50000
100000
150000
200000
250000
300000


Satisfied Users
Dissatisfied Users
Frequency
Cumulative_Lim_Service_Time_LTE
0
10
20
30
40
50
60
80
100
1000
2000
3000
4000
5000
6000
7000



Satisfied Users
Dissatisfied Users
Frequency
Cumulative_No_Service_Time_LTE
0
10
20
30
40
50
60
70
80
100
120
140
160
2000
4000
6000
8000
10000
12000
14000

Interpretation of Histograms

- As evident from our dataset's histograms, the majority of our features exhibit asymmetric distributions, such as Bimodal or Skewed shapes.
- Only a few features closely resemble a Normal Distribution.
- To effectively model our data, we should prioritize classification techniques that are well-suited for asymmetric data.
- This is our motivation why we used Decision Trees, Random Forests, Gradient Boosting Models, as they are known for their robustness in handling such data distributions.

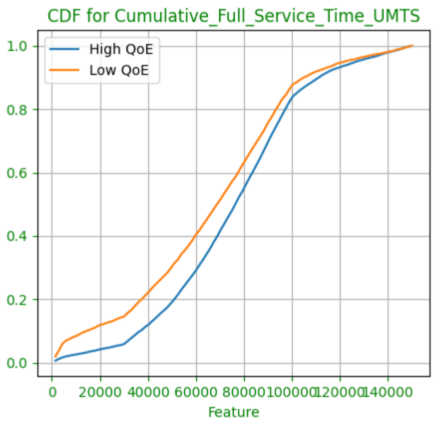
4. Feature Analysis

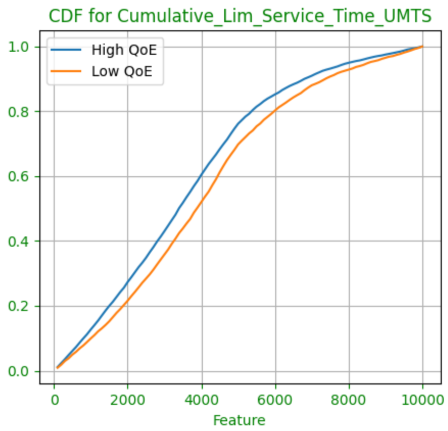
4.2. Cumulative distribution function (CDF): The features of dataset 1 are sorted based on the highest gap between satisfied and unsatisfied users in CFD plots.

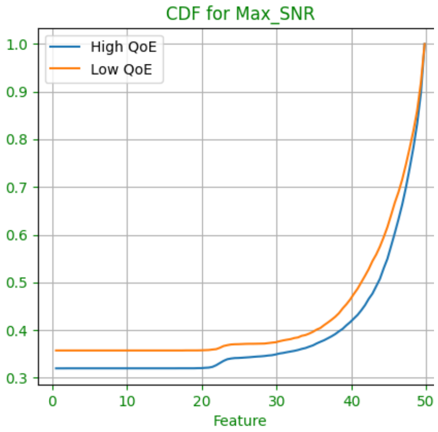
Feature Name	Gap Value
1. Cumulative_Full_Service_Time_UMTS :	0.1136
2. Cumulative_Lim_Service_Time_UMTS :	0.0852
3. Max_SNR:	0.0683
4. Cumulative_Full_Service_Time_LTE :	0.0507
5. Max_RSRQ :	0.0451
6. Cumulative_Lim_Service_Time_LTE :	0.0443
7. Cumulative_YoutubeSess_LTE_DL_Volume :	0.0257
8. Cumulative_No_Service_Time_UMTS :	0.0225
9. Cumulative_No_Service_Time_LTE :	0.0197
10. Cumulative_YoutubeSess_UMTS_DL_Time :	0.0074
11. Cumulative_YoutubeSess_LTE_DL_Time :	0.0065
12. Cumulative_YoutubeSess_UMTS_DL_Volume :	0.0044

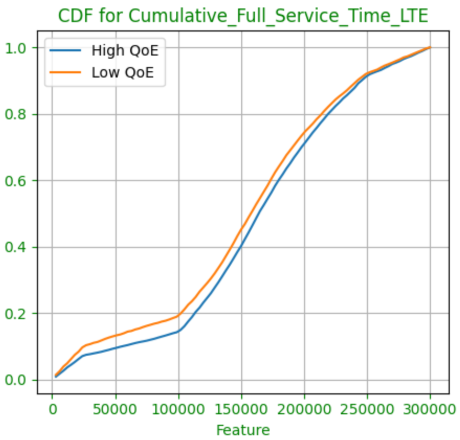
4.2. Relation between user QoE and data distribution - dataset 1: CDF

The CDF plots of dataset 1 are sorted based on the highest gap between satisfied and unsatisfied users (left to right).

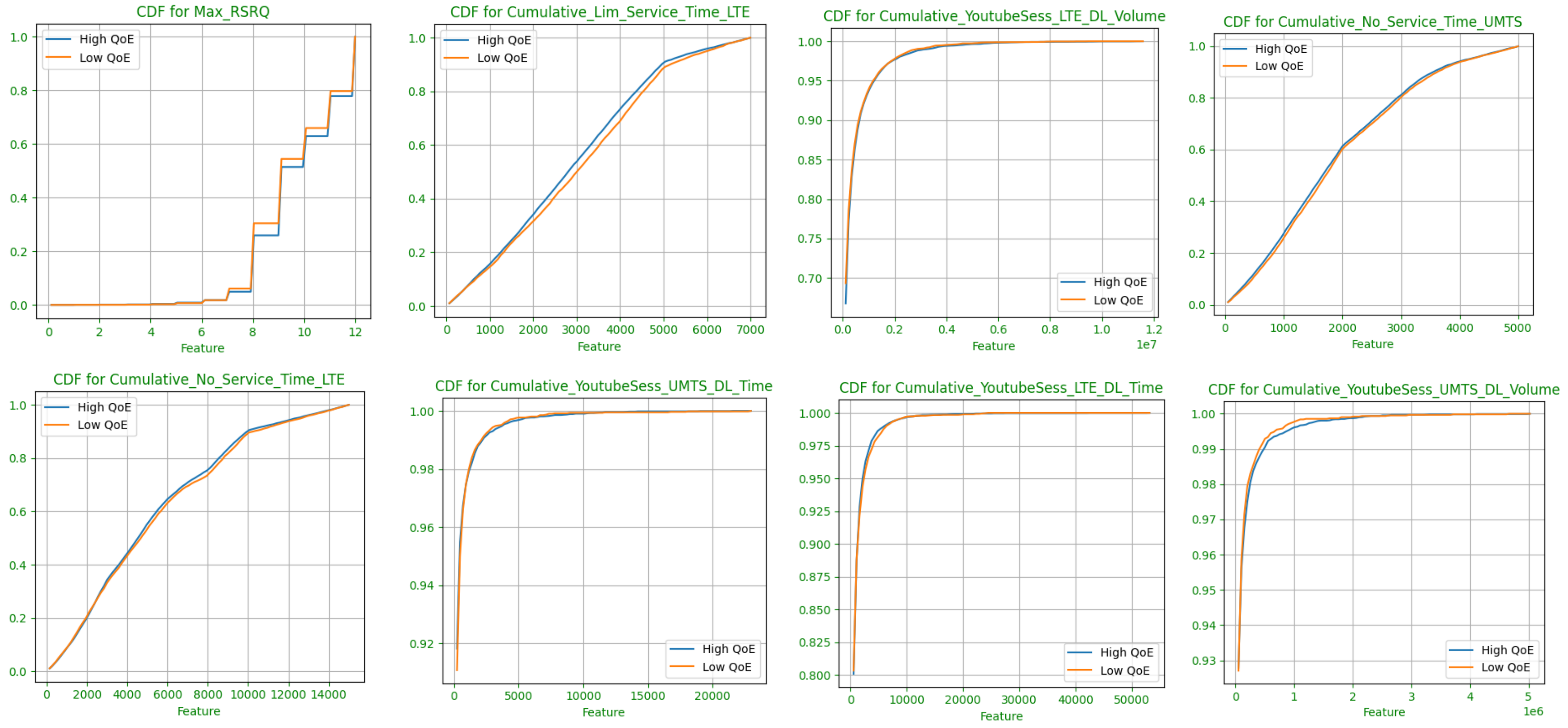








4.2. Relation between user QoE and data distribution - dataset 1: CDF

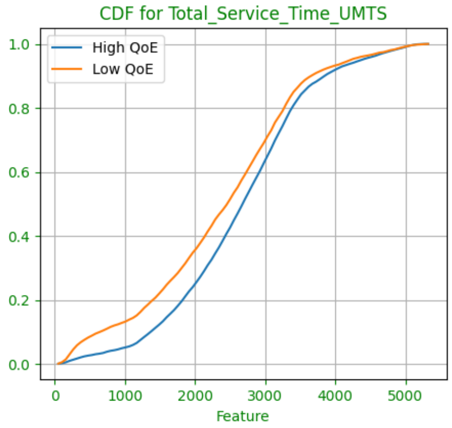


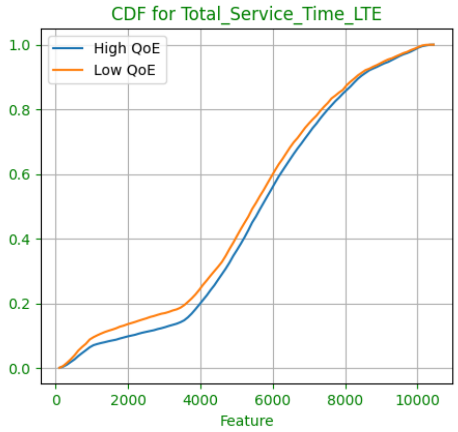
The features of dataset 2 are sorted based on the highest gap between satisfied and unsatisfied users in CFD plots.

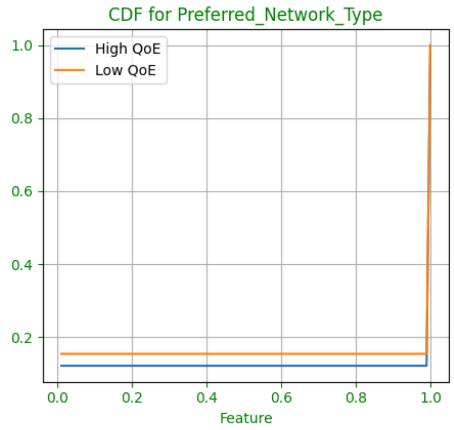
	Feature Name	Gap Value
1.	Total_Service_Time_UMTS :	0.1072
2.	Total_Service_Time_LTE :	0.0477
3.	Preferred_Network_Type :	0.0326
4.	Avg_Youtube_Volume_LTE :	0.0257
5.	Youtube_Time_Percentage_LTE :	0.0257
6.	Service_Availability_Ratio_LTE :	0.0175
7.	Youtube_Time_Percentage_UMTS :	0.0124
8.	Avg_Youtube_Time_UMTS :	0.0074
9.	Avg_Youtube_Time_LTE :	0.0065
10.	Avg_Youtube_Volume_UMTS :	0.0044
11.	Service_Availability_Ratio_UMTS :	0.0044

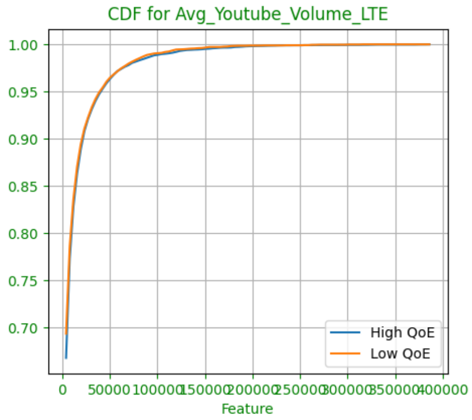
4.2. Relation between user QoE and data distribution - dataset 2: CDF

The CDF plots of dataset 2 are sorted based on the highest gap between satisfied and unsatisfied users (left to right).

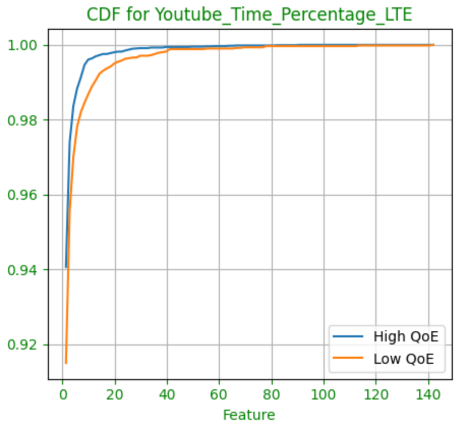


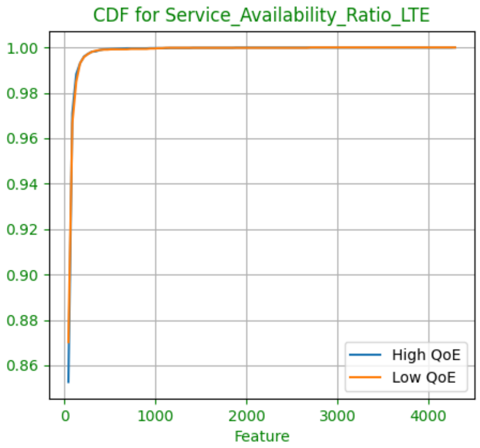


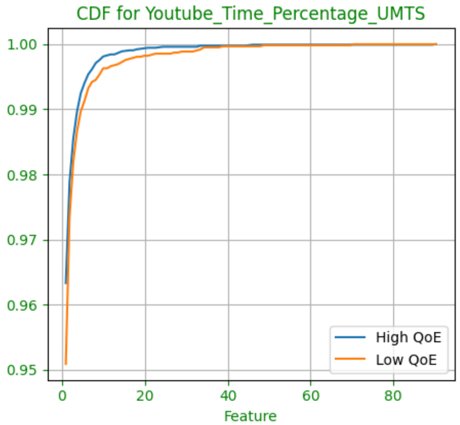


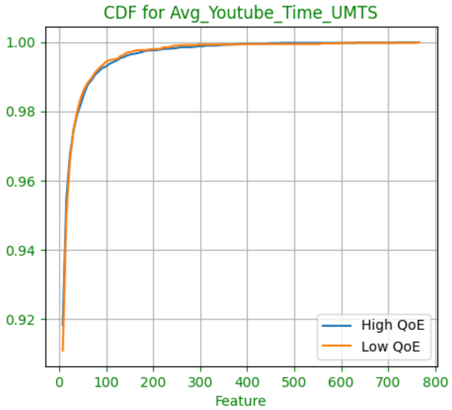


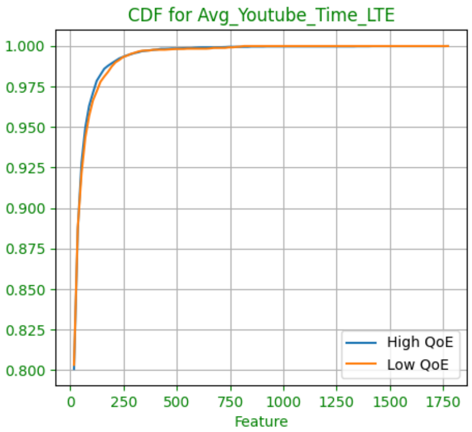
4.2. Relation between user QoE and data distribution - dataset 2: CDF

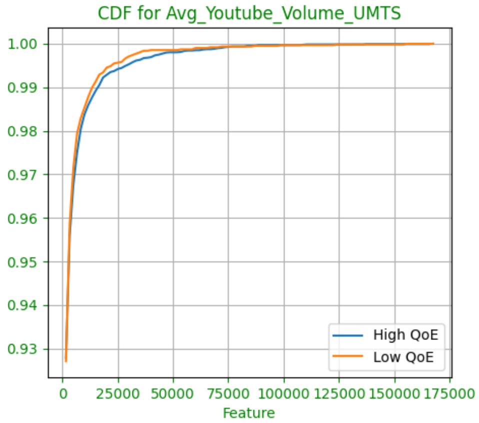


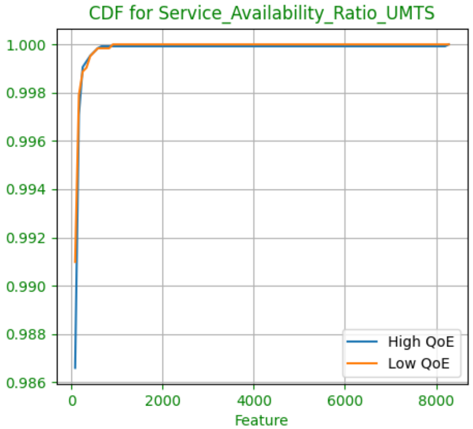












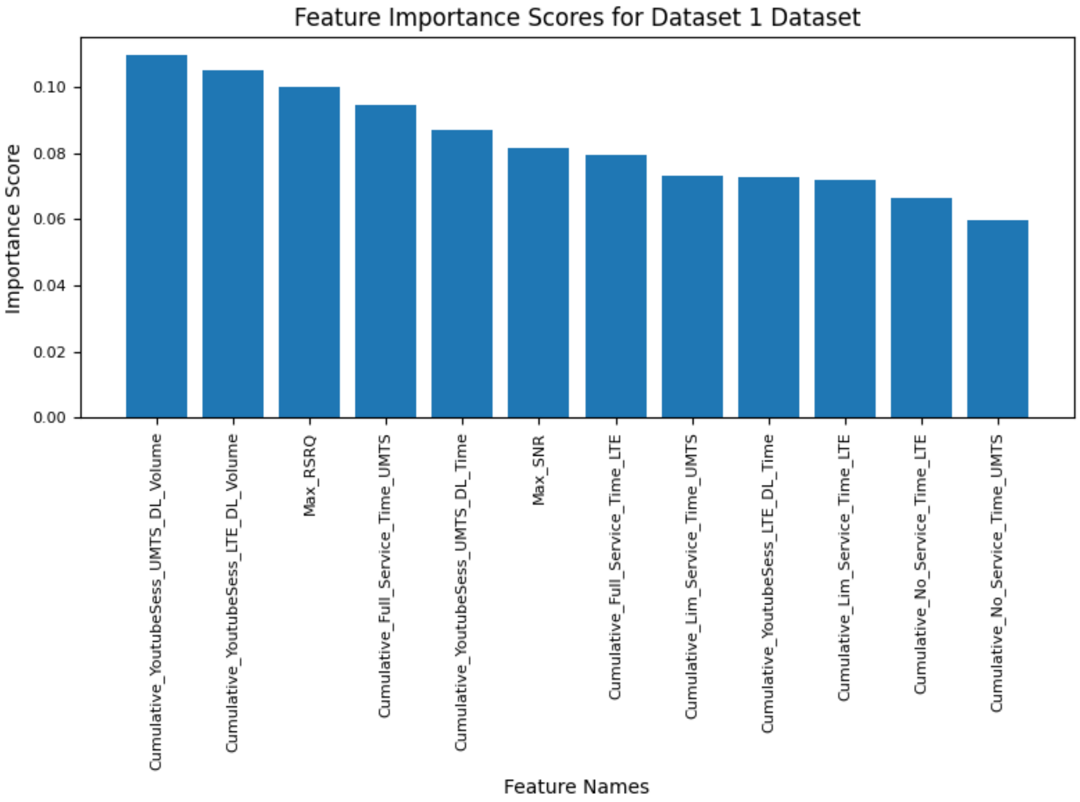
The features of dataset 3 are sorted based on the highest gap between satisfied and unsatisfied users in CFD plots.

Feature Name	Gap Value
1. Cumulative_Full_Service_Time_UMTS :	0.1136
2. Total_Service_Time_UMTS :	0.1072
3. Cumulative_Lim_Service_Time_UMTS :	0.0852
4. Max_SNR :	0.0683
5. Cumulative_Full_Service_Time_LTE :	0.0507
6. Total_Service_Time_LTE :	0.0477
7. Max_RSRQ :	0.0451
8. Cumulative_Lim_Service_Time_LTE :	0.0443
9. Preferred_Network_Type :	0.0326
10. Cumulative_YoutubeSess_LTE_DL_Volume :	0.0257
11. Avg_Youtube_Volume_LTE :	0.0257
12. Youtube_Time_Percentage_LTE :	0.0257
13. Cumulative_No_Service_Time_UMTS :	0.0225
14. Cumulative_No_Service_Time_LTE :	0.0197

Feature Name	Gap Value
15. Service_Availability_Ratio_LTE :	0.0175
16. Youtube_Time_Percentage_UMTS :	0.0124
17. Cumulative_YoutubeSess_UMTS_DL_Time :	0.0074
18. Avg_Youtube_Time_UMTS :	0.0074
19. Cumulative_YoutubeSess_LTE_DL_Time :	0.0065
20. Avg_Youtube_Time_LTE :	0.0065
21. Cumulative_YoutubeSess_UMTS_DL_Volume :	0.0044
22. Avg_Youtube_Volume_UMTS :	0.0044
23. Service_Availability_Ratio_UMTS :	0.0044

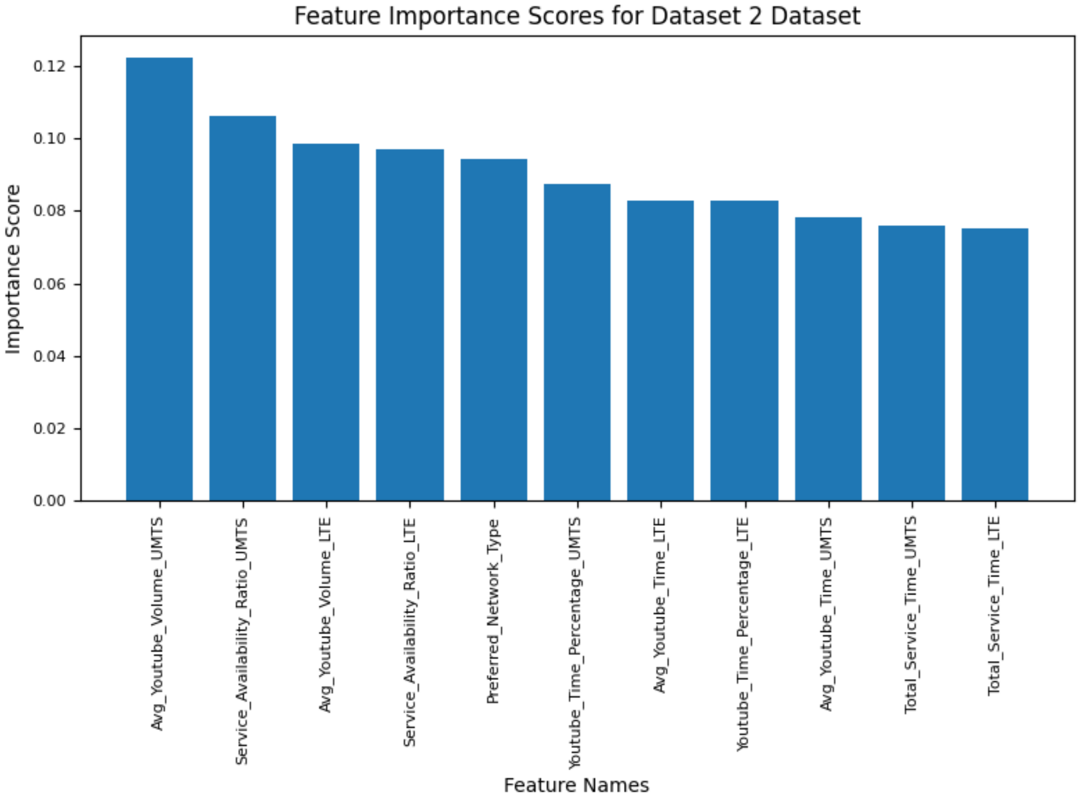
4. Feature Analysis

4.3. XGBoost + Bayesian optimization hyperparameter tuning - Dataset 1



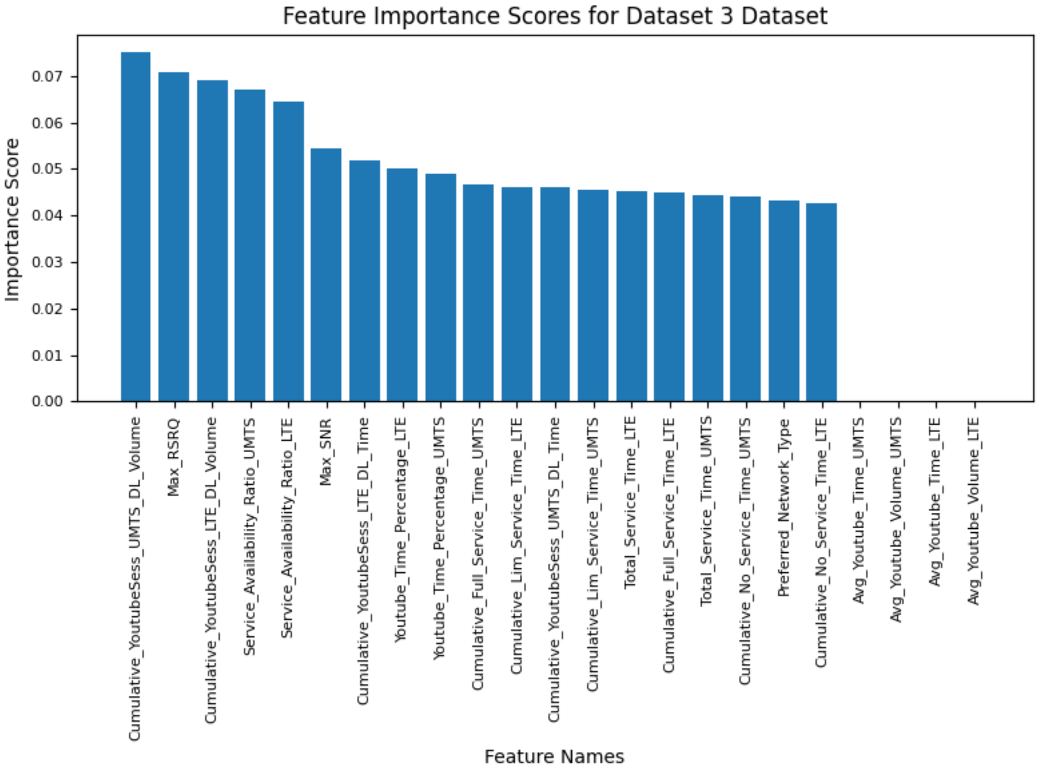
Observation: Across all features, their importance scores fall within the range of 0.06 to 0.10. This indicates that XGBoost recognizes these features as more informative compared to CDF. Interestingly, the features with the highest importance scores in XGBoost show lower significance in CDF, and vice versa.

4.3. XGBoost + Bayesian optimization hyperparameter tuning - Dataset 2



Observation: The same thing as before also happens here. For example,Avg_Youtube_Volume_UMTS is the most important feature in dataset 2 according to XGBoost, while it is the second-to-last feature according to CDF.

4.3. XGBoost + Bayesian optimization hyperparameter tuning - Dataset 3



Observation: As illustrated in the chart, in this scenario, the features of dataset 1 appear to be more informative when compared to dataset 2.

5. Model Deployment

Classifiers and Hyperparameters:

- Decision Tree: Max depth: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10
- Random Forest: # of estimators: 20, 40, 50 | Max depth: 10, 20 ,30
- XGBoost: # of estimators: 35, 45, 55 | Max depth: 1, 10, 20 | learning rate 0.001, 0.005, 0.01

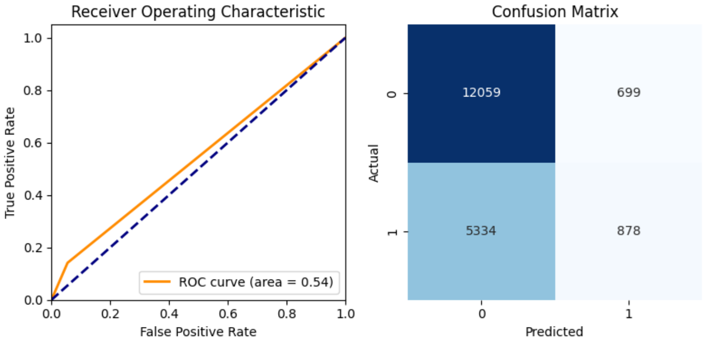
Cross-Validation: 5 fold

Normalization: Standard scaler

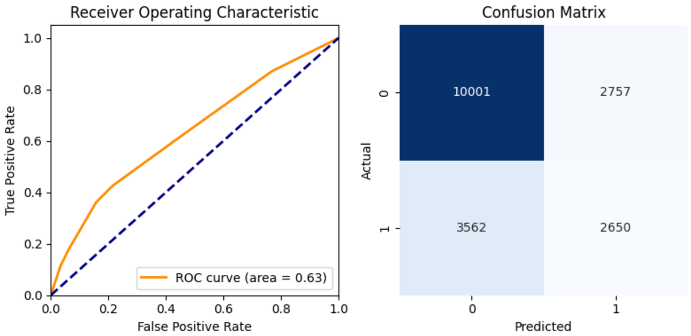
Hyperparameter Tuning: GridSearch

6.1.Training Results
6.1.1. Decision Tree

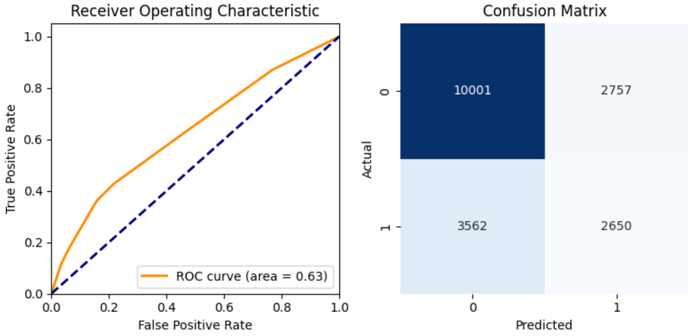
<u>Dataset 1</u>
Time taken: 5.2 seconds
Best Hyperparameters: {'max_depth': 1}



<u>Dataset 2</u>
Time taken: 5.2 seconds
Best Hyperparameters: {'max_depth': 3}

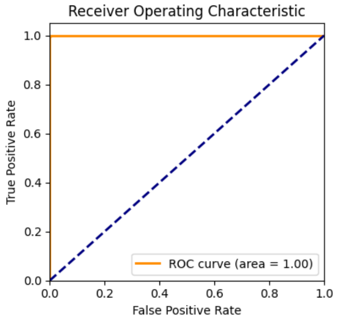


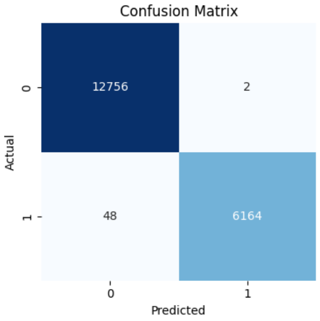
<u>Dataset 3</u>
Time taken: 10.5 seconds
Best Hyperparameters: {'max_depth': 3}



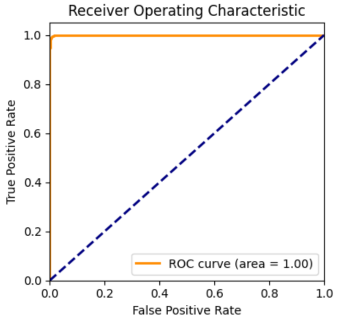
6.1.2. Random Forest

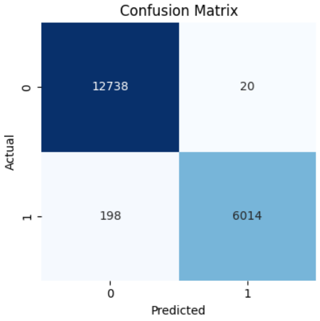
<u>Dataset 1</u>
Time taken: 1 minutes and 9.8 seconds
Best Hyperparameters:
{'max_depth': 20, 'n_estimators': 50}



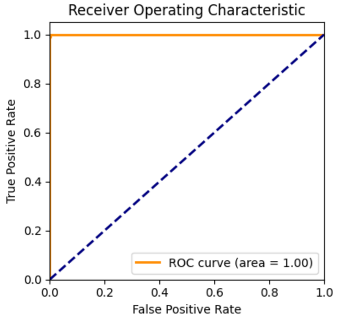


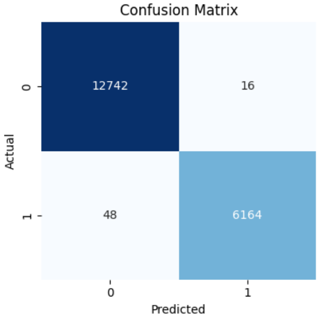
<u>Dataset 2</u>
Time taken: 1 minutes and 11.0 seconds
Best Hyperparameters:
{'max_depth': 20, 'n_estimators': 50}





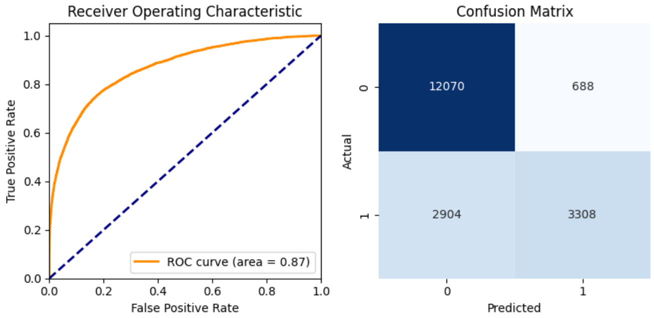
<u>Dataset 3</u>
Time taken: 1 minutes and 31.9 seconds
Best Hyperparameters:
{'max_depth': 20, 'n_estimators': 50}



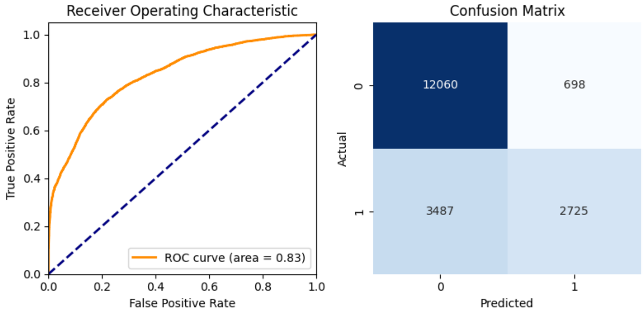


6.1.3. XGBoost

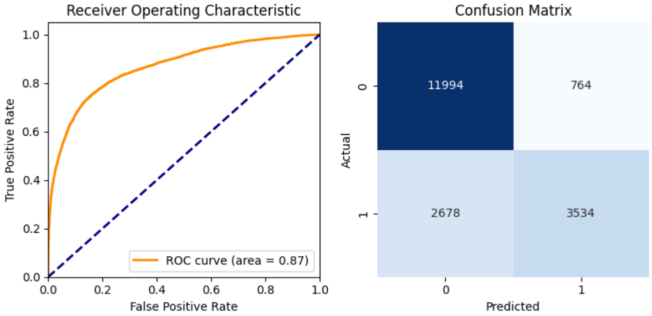
<u>Dataset 1</u>
Time taken: 5 minutes and 52.2 seconds
Best Hyperparameters:
{'learning_rate': 0.01, 'max_depth': 10, 'n_estimators': 55}



<u>Dataset 2</u>
Time taken: 5 minutes and 29.5 seconds
Best Hyperparameters:
{'learning_rate': 0.01, 'max_depth': 10, 'n_estimators': 55}



<u>Dataset 3</u>
Time taken: 10 minutes and 2.7 seconds
Best Hyperparameters:
{'learning_rate': 0.005, 'max_depth': 10, 'n_estimators': 55}



6.2. Testing Results
6.2.1. Decision Tree

Observations:

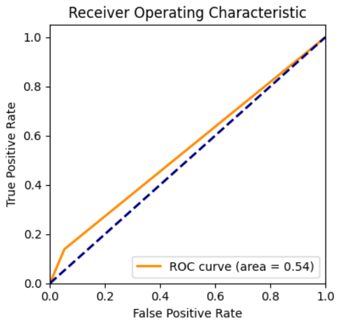
- True Negatives:
Dataset 1 > Dataset 2 = Dataset 3

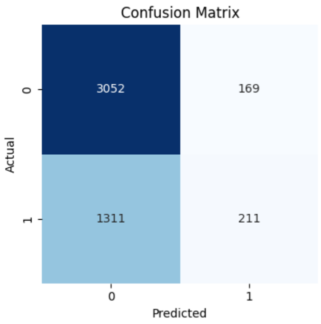
- True Positives:
Dataset 2 = Dataset 3 > Dataset 1

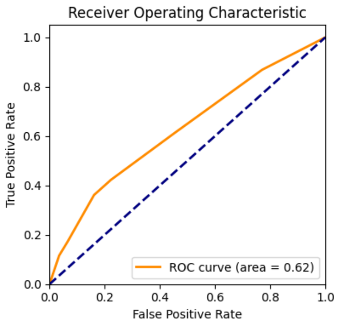
- False Negatives:
Dataset 1 > Dataset 2 = Dataset 3

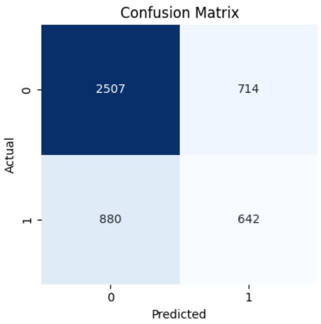
- False Positives:
Dataset 2 = Dataset 3 > Dataset 1

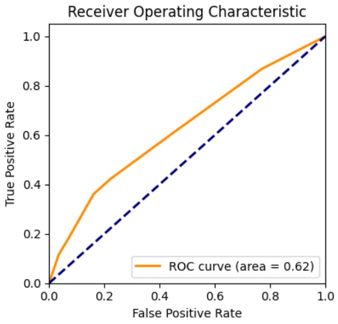
- AUC Value
Dataset 2 = Dataset 3 > Dataset 1

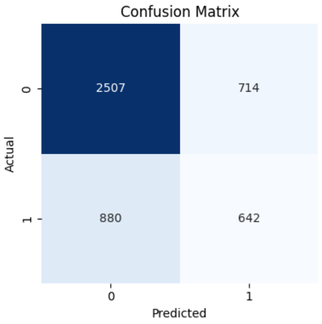












6.2.2. Random Forest

Observations:

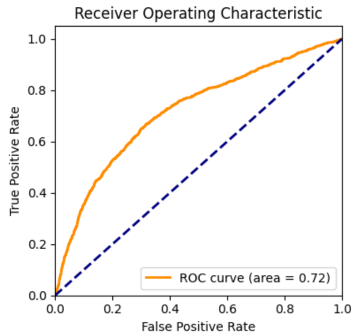
- True Negatives:
Dataset 1 > Dataset 3 > Dataset 2

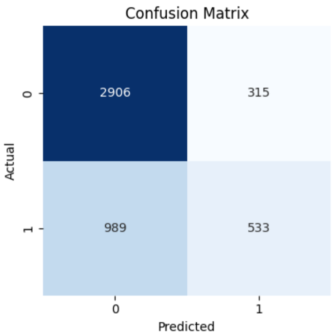
- True Positives:
Dataset 3 > Dataset 1 > Dataset 2

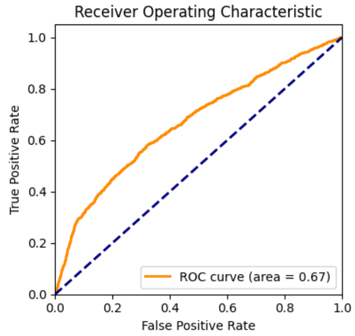
- False Negatives:
Dataset 2 > Dataset 1 > Dataset 3

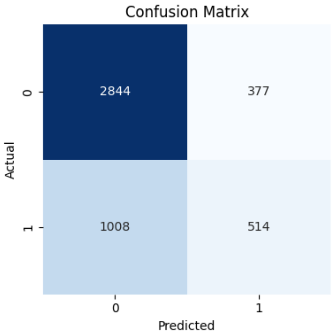
- False Positives:
Dataset 2 > Dataset 3 > Dataset 1

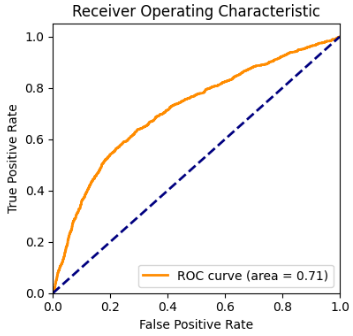
- AUC Value
Dataset 1 > Dataset 3 > Dataset 2

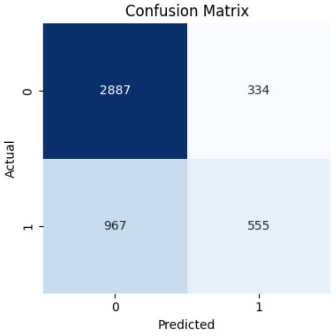












6.2.3. XGBoost

Observations:

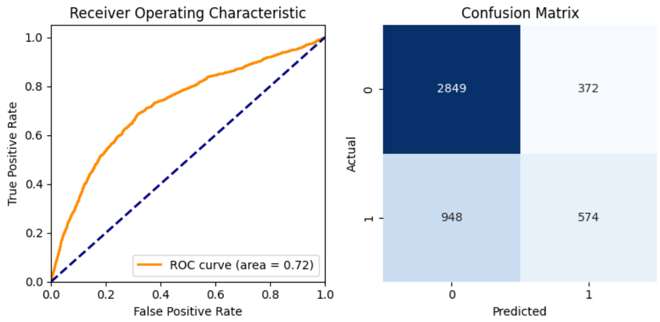
- True Negatives:
Dataset 2 > Dataset 1 > Dataset 3

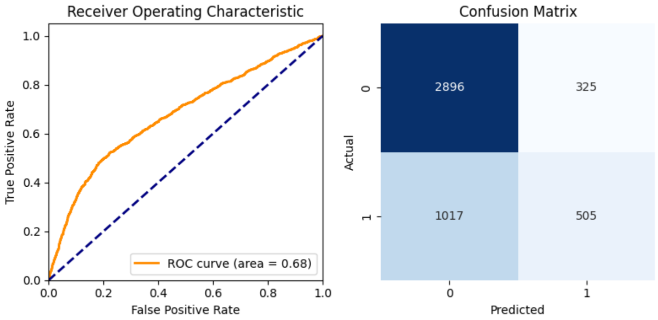
- True Positives:
Dataset 3 > Dataset 2 > Dataset 1

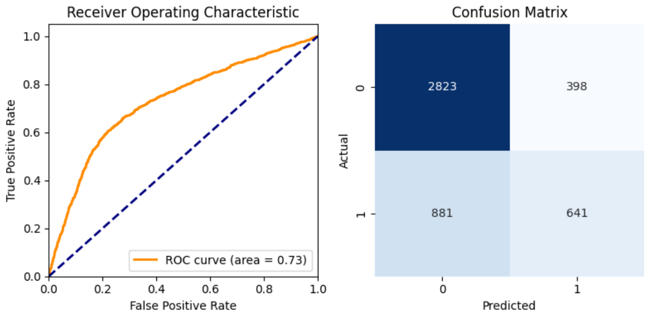
- False Negatives:
Dataset 2 > Dataset 1 > Dataset 3

- False Positives:
Dataset 3 > Dataset 1 > Dataset 2

-AUC Value
Dataset 3 > Dataset 1 > Dataset 2







7. Comparison

Algorithm	Dataset	Training AUC	Testing AUC	Time
DT	1	0.54	0.54	**5.2 s**
	2	0.63	0.62	5.2 s
	3	0.63	0.62	10.5 s
RF	1	**1**	0.72	9.8 s
	2	1	0.67	11.0 s
	3	1	0.71	31.9 s
XGBoost	1	0.87	0.72	5 m, 52.2 s
	2	0.83	0.68	5 m, 29.5 s
	3	0.87	**0.73**	10 m, 2.7 s

8. Conclusion

- Given that we are not dealing with expensive-to-evaluate functions in the training process, Grid Search emerges as a swifter and more straightforward approach compared to Bayesian Optimization.
- Regarding feature ranking outcomes from cumulative distribution functions (CDF) and XGBoost, the greater reliability of XGBoost in this task can be attributed to its robustness and adaptability.
- When considering execution time, the order is XGBoost > Random Forest (RF) > Decision Tree (DT).
- In the context of training AUC (Area Under the Curve), RF surpasses XGBoost, followed by DT. In terms of test AUC, XGBoost, and RF perform equally well, outperforming DT.
- Notably, Dataset 3 exhibits the most favorable results among the datasets considered.

9. References

[1] https://scikit-learn.org/stable/modules/tree.html#

[2] https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html#

[3] https://xgboost.readthedocs.io/en/stable/

[4] https://scikit-learn.org/stable/modules/grid_search.html#

[5] https://distill.pub/2020/bayesian-optimization/