5G ML Challenge:

Classification of Home Network Users to Improve User Experience

Team: Spears-9

Hayden Myers (A11732447), Chhavi Nijhawan (A20083323),

Sushma Reddy (A20348652), Anuj Singh (A20289993), Kumar Yash (A20346522)

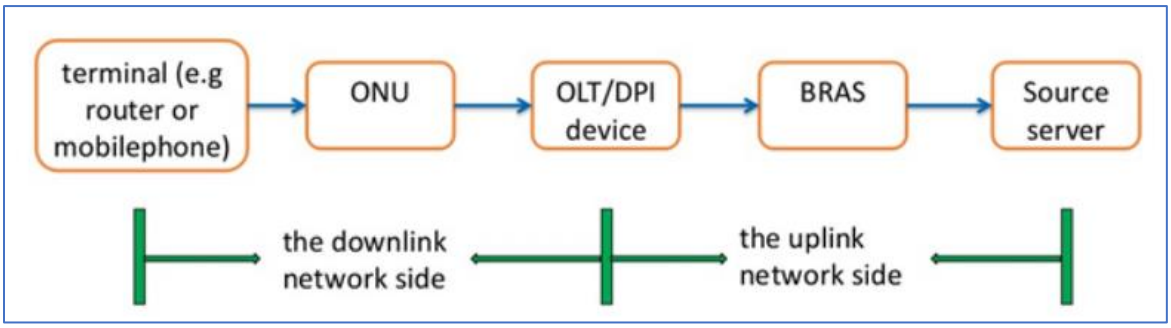
Table of Contents

Business Understanding	3
Data Understanding	4
Data Preparation and Preprocessing	5
Data Cleaning and Outlier Detection	5
Data Transformation	7
Data Modeling	9
Model Evaluation	13
Appendix	13
References	22

Business Understanding

With the ever-changing and rapid growth of technology with mobile phones and the Internet, the home broadband system has become a key part of many people around the world's day-to-day life. This increase has led to a progressively saturated market which is dependent upon the user's experience and the quality of broadband networks to determine whether the market is competitive with the users at hand.

Many factors affect the user's online experience. The DPI probe divides the broadband end-to-end network into two parts: the uplink network side and the downlink network side. The network layout is shown in the below figure. Downlink network side problems account for a large proportion of network problems. Many operators have turned their attention to detecting problems with the network's quality and improving the user's experience as fast as possible. Accurately depicting whether a user is truly experiencing something classified as "good" or "bad" is a big concern for the operators in the future so they may be able to discover a complaint from the user before it ever happens.



Data Understanding

The data we were provided contained 8 key indicators which are listed below. The indicators are obtained from the three-way handshake process and the data transmission process after the handshake is successful. The specific physical meanings of the indicators are as follows:

1. **Indicator 1:** In the first step of the three-way handshake, the time interval between the syn ack packet and the ack packet.
2. **Indicator 2:** In the second step of the three-way handshake, the time interval between the syn ack packet and the ack packet.
3. **Indicator 3:** The time interval between the ack packet and the first payload packet in the three-way handshake.
4. **Indicator 4:** The response delay of the first packet with payload after the establishment of TCP for multiple flows in the session.
5. **Indicator 5:** In TCP transmission, the actual delay of transmission from the DPI position to the user terminal.
6. **Indicator 6:** In TCP transmission, the transmission delay from the DPI position to the website.
7. **Indicator 7:** In TCP transmission, the percentage of downlink retransmitted packets in the current session.
8. **Indicator 8:** In TCP transmission, the percentage of upstream retransmission packets of the current session.

Additionally, the data that was given to us included the day, hour, specific time, and the 8 indicator variables for each user on a good or bad user experience basis. The user's good experience (UGE) and the user's bad experience (UBE) are in different files. These experiences are what we as a team are trying to accurately predict for future cases. We began looking into the data and have found that for certain specific times, a user can experience multiple things and have a good or bad experience in the same timestamp. This is something to keep in mind as we move further along with the data cleaning and preparation for modeling to be done appropriately.

Data Preparation and Preprocessing

The data we were provided has one CSV file for each user. There were 50 users with a good user experience (UGE) and 50 with a bad experience (UBE) within the validation and test set. Within the training set, we were given 150 UGE and 150 UBE. For each user in the data, they collected multiple timestamps for a week-long period (6/10/2021 - 6/16/2021).

We merged all the training users into one file, the test users into one file, and validation into another. After doing so, we created a variable named "ID" to identify each user throughout the data we were provided. Next, we added the user experience type as a binary variable named "Type". We gave the values to the user's experience, 0 for the UBE, and then the UGE was given the value of 1.

We created multiple different files. In the first file we created, we aggregated the data by the individual timestamp and took the mean of all of the records for the specific timestamp. We did the same thing again but on an hourly basis. Last, we aggregated the data with one row for each user and made that a record. Since the user data was relatively dense, hence we tried different kinds of aggregations to feed classification models with different versions of preprocessed data. We also did the same process again but changed the aggregation to the median instead of the mean. Each of these files was made for test, training, and validation sets which brought us to a total of 18 files.

It is always important to check for duplicates in the data; however, in our case, we did not find any duplicates of data and did not have to remove any data due to this common issue.

Data Cleaning and Outlier Detection

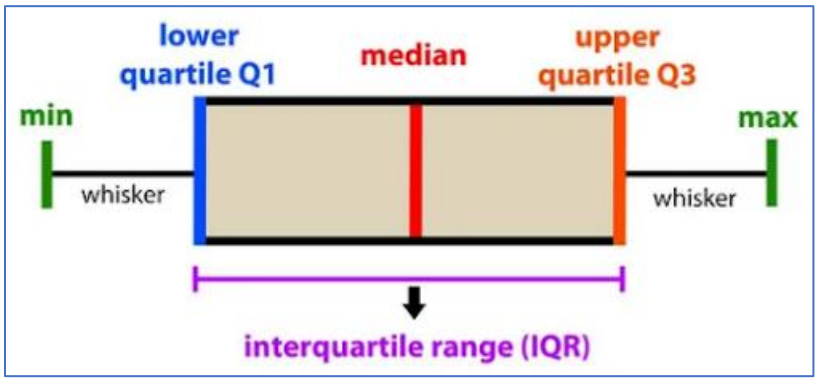
Min-Max Scaling:

We tried the min-max scaling method to deal with the outliers. As part of this, the scalar takes each value and subtracts the minimum, and then divides by the range. The resultant values range

between 0 and 1. The reason behind using this approach is to reduce the standard deviation to minimize the effect of outliers on the model's performance.

IQR Method:

We used IQR methods to detect the outliers and remove them from the data as part of data cleaning. To explain it further, take the below picture as a reference:



This box plot above gives an idea about the current distribution of raw data. The 'minimum' in the above box plot represents our dataset's minimum value, and the same goes for 'maximum'. The difference between the maximum and minimum gives us the range of the data provided.

Q1 is the 4th percentile in our case with this data, which means 4 percent of the data lies between the minimum and Q1. The 'median' represents the center point or second quartile of the data. Q3 is the 96th percentile and represents 96 percent of the data that lies between the minimum and Q3. The interquartile range (IQR) is the difference between Q3 and Q1.

To find the outlier using the IQR method we should define a new range called the "decision range", thus any data point out of this range is considered an outlier.

Lower Bound = (Q1 - 1.5*IQR)

Upper Bound = (Q3 + 1.5*IQR)

The data points that were lower than the "Lower Bound" or higher than the "Upper Bound" were considered outliers and removed from the data.

Data Transformation

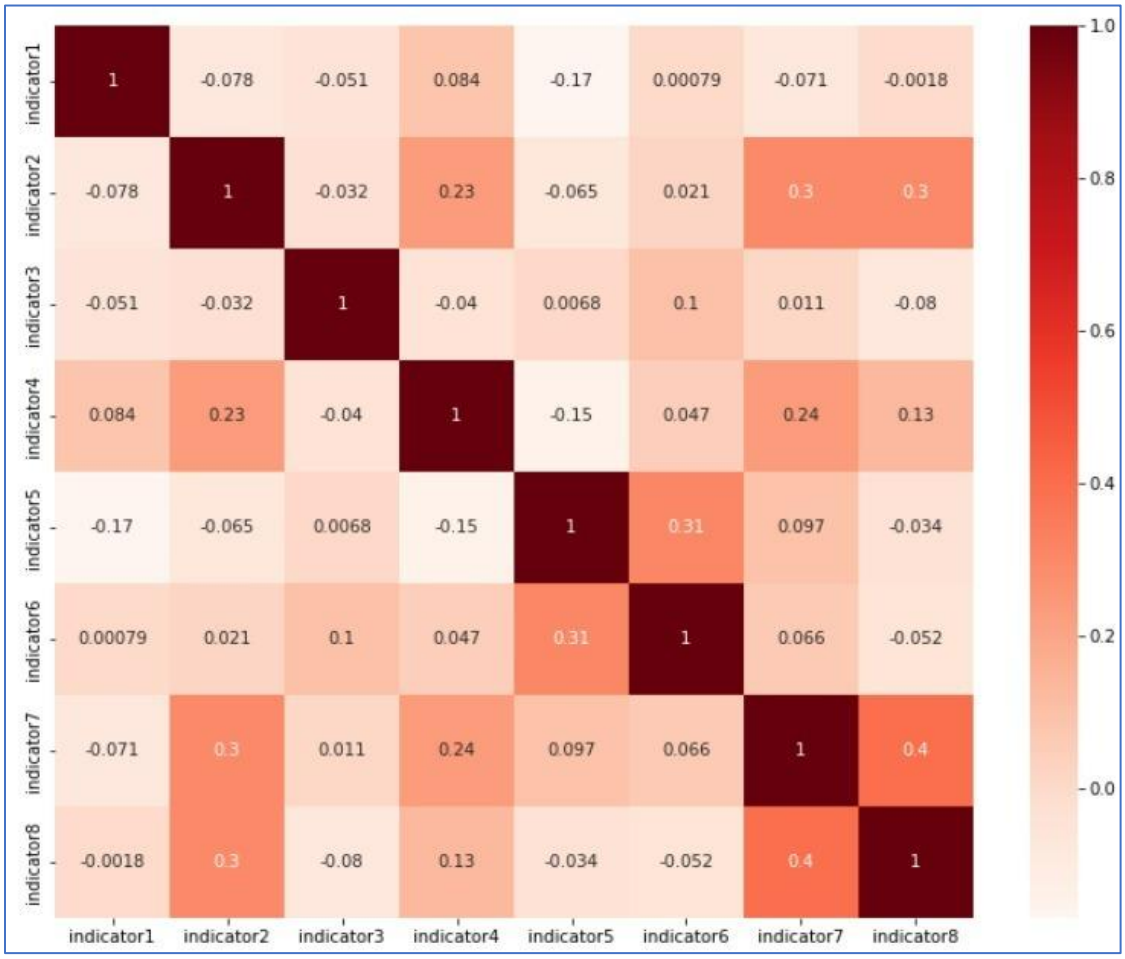
We aggregated all the recordings of each user after removing outliers from the data, below we have records of 150 good users and 150 bad users.

The below picture gives us the statistics of users before data cleaning and after cleaning:

	indicator1	indicator2	indicator3	indicator4	indicator5	indicator6	indicator7	indicator8
count	3687372.00	3687372.00	3687372.00	3687372.00	3687372.00	3687372.00	3687372.00	3687372.00
mean	81.58	189.42	135.84	164.48	47.46	21.82	0.02	0.02
std	1526.90	2037.41	1715.12	2925.86	100.90	321.15	0.05	0.07
min	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
25%	4.40	3.00	0.64	4.50	5.00	8.00	0.00	0.00
50%	11.79	6.89	2.58	12.75	19.00	15.00	0.00	0.00
75%	22.50	39.00	8.22	24.00	49.00	25.00	0.01	0.00
max	487451.50	303685.00	432725.00	575323.00	2000.00	362002.00	0.50	0.50

	indicator1	indicator2	indicator3	indicator4	indicator5	indicator6	indicator7	indicator8
count	3617140.00	3461103.00	3485637.00	3587911.00	3599378.00	3651024.00	3574728.00	3406341.00
mean	14.08	26.94	6.68	15.32	35.43	17.42	0.01	0.00
std	12.22	52.33	12.05	13.61	46.73	13.01	0.03	0.01
min	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
25%	4.30	2.50	0.50	4.50	5.00	8.00	0.00	0.00
50%	11.50	6.00	2.22	12.14	17.00	15.00	0.00	0.00
75%	21.78	33.00	6.50	23.00	46.00	25.00	0.01	0.00
max	79.15	371.25	82.50	93.69	272.00	85.00	0.18	0.05

Covariance Matrix of Indicators:



Data Modeling

We plotted learning curve plots for our model which states the score over varying numbers of training samples, while a validation curve plots the score over a varying hyperparameter.

A learning curve is a tool for finding out if an estimator would benefit from more data, or if the model is too simple (biased).

Logistic Regression:

Logistic Regression is often used for classification and predictive analytics. Based on a given dataset of independent variables, logistic regression calculates the likelihood that an event will occur, such as voting or not voting. Given that the result is a probability, the dependent variable's range is 0 to 1. In logistic regression, the odds—that is, the probability of success divided by the probability of failure—are transformed using the logit formula. The following formulas are used to represent this logistic function, which is sometimes referred to as the log odds or the natural logarithm of odds:

$$Logit(pi) = 1/(1+ \exp(-pi))$$

$$\ln(pi/(1-pi)) = Beta\_0 + Beta\_1*X\_1 + \ldots + B\_k*K\_k$$

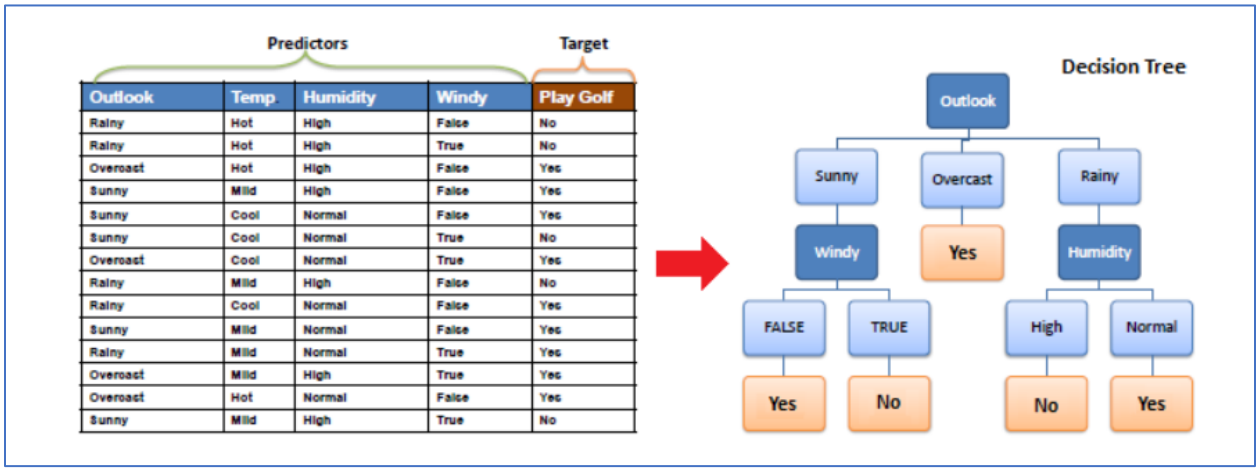
The evaluated model accuracy for the Logistic Regression model is 57% as per Appendix Image 1

Decision Tree Classification:

Using a tree structure, decision trees construct classification or regression models. It incrementally develops an associated decision tree while segmenting a dataset into smaller and smaller sections. The outcome is a tree containing leaf nodes and decision nodes. Outlook is one example of a decision node that has two or more branches (e.g., Sunny, Overcast, and Rainy). A classification or choice is represented by a leaf node (for instance, Play). The root node is the topmost decision

node in a tree and corresponds to the best predictor. Both category and numerical data can be processed using decision trees.

The evaluated model accuracy for the Decision Tree classification model is 47% as per Appendix Image 2



Random Forest Classification:

Supervised machine learning algorithms, like the random forest, are frequently employed in classification and regression issues. On various samples, it constructs decision trees and uses their average for classification and majority vote for regression. The Random Forest Algorithm's ability to handle data sets with both continuous variables, as in regression, and categorical variables, as in classification, is one of its most crucial qualities. In terms of classification issues, it delivers superior outcomes.

The evaluated model accuracy for the Random Forest classification model is at 44% as per Appendix Image 3

XG Boost:

XG Boost which stands for Extreme Gradient Boosting is a library that focuses on computational speed and model performance. It supports Gradient boosting, Stochastic Gradient Boosting, and Regularized Gradient Boosting.

The gradient-boosted trees approach is widely used and well-implemented in open-source software called XGBoost. Gradient boosting is a supervised learning process that combines the predictions of several weaker, simpler models to attempt to properly predict a target variable.

The evaluated model accuracy for the XG Boost model is at 57% as per Appendix Image 4

AdaBoost Algorithm:

AdaBoost also known as Adaptive Boosting is a Machine Learning algorithm used as an Ensemble Method. The most common AdaBoost algorithm is decision trees with one level, which means decision trees with only one split. These trees are also referred to as Decision Stumps. This algorithm constructs a model and assigns equal weights to all data points. It then assigns higher weights to incorrectly classified points. In the following model, all points with higher weights are given more weight. It will continue to train models until a lower error is received.

The evaluated model accuracy for the AdaBoost model is at 57% as per Appendix Image 5

Extra Tree Classifier Algorithm:

Extra Tree Classifier is a type of ensemble learning technique that outputs a classification result by aggregating the results of multiple de-correlated decision trees collected in a "forest." In concept, it is very similar to a Random Forest Classifier and differs only in the way the decision trees in the forest are constructed. The Extra Tree Forest's Decision Trees are built from the original training sample. Then, at each test node, each tree is given a random sample of k features from the feature set, from which each decision tree must choose the best feature to split the data using some mathematical criteria (typically the Gini Index). This random selection of features results in the construction of multiple de-correlated decision trees.

The evaluated model accuracy for the Extra Tree model is at 45% as per Appendix Image 6

KNN:

The K-Nearest Neighbors (KNN) was used to predict whether a user was having a good or bad experience based on the 8 key indicators. This algorithm is a supervised machine learning algorithm.

KNN finds the distances between a query and each example in the data, chooses the K examples closest to the query, and then, in the case of classification, votes for the label with the highest frequency or averages the labels (in the case of regression).

The evaluated model accuracy for the K-Nearest Neighbors model is at 46% as per Appendix Image 7

Support Vector Classification:

A support vector machine (SVM) is a supervised machine learning model that solves two-group classification problems using classification techniques. An SVM model can classify new text after being given sets of labeled training data for each category.

They offer two key advantages over more recent algorithms like neural networks: greater speed and improved performance with fewer samples (in the thousands). As a result, the approach is excellent for text classification issues, where it's typical to only have access to a dataset with a few thousand tags on each sample.

The evaluated model accuracy for the Support Vector model is at 51% as per Appendix Image 8

Gaussian Naive Bayes:

Gaussian Naive Bayes is a probabilistic classification algorithm that uses the Bayes theorem with strong independence assumptions. The concept of independence in classification refers to the idea that the presence of one value of a feature does not influence the presence of another (unlike independence in probability theory).

The evaluated model accuracy for the Gaussian Naive Bayes model is at 52% as per Appendix Image 9

Model Evaluation

Classification Model Name	Model Accuracy %
Logistic Regression	57
Decision Tree Classification	47
Random Forest Classification	44
Gradient Boosting Classification	57
Ada Boosting Classification	56
Extra Tree Classification	45
K-Neighbors Classification	46
Support Vector Classification	51
Gaussian Naive Bayes	52

Appendix

Image 1:

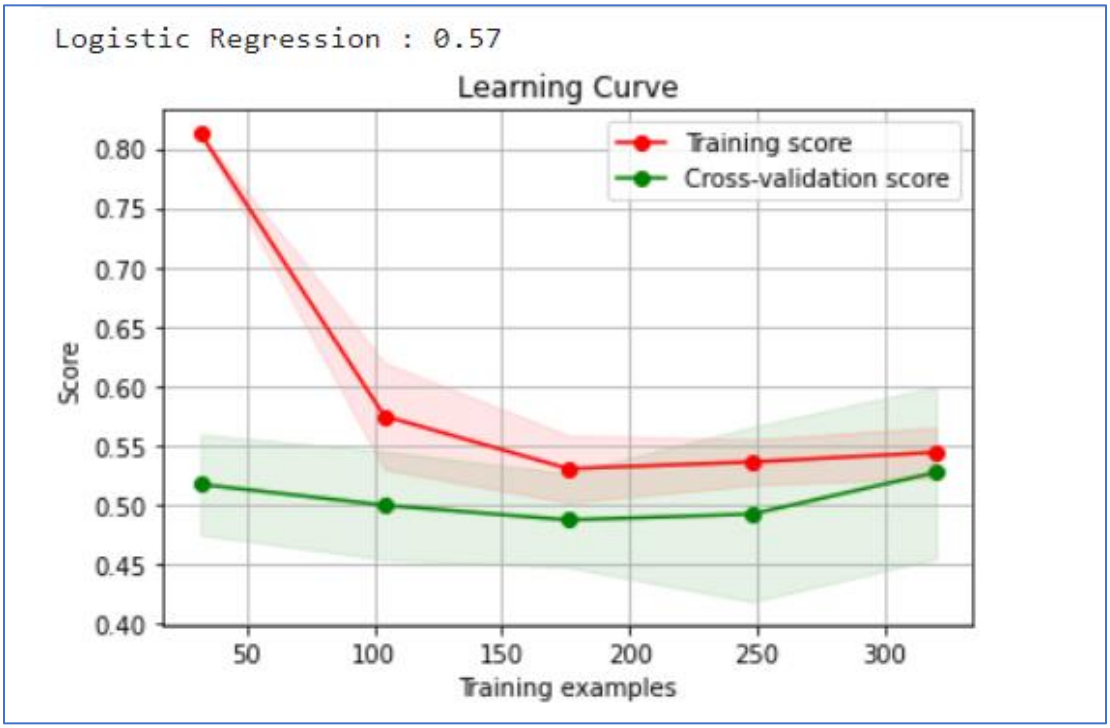


Image 2:

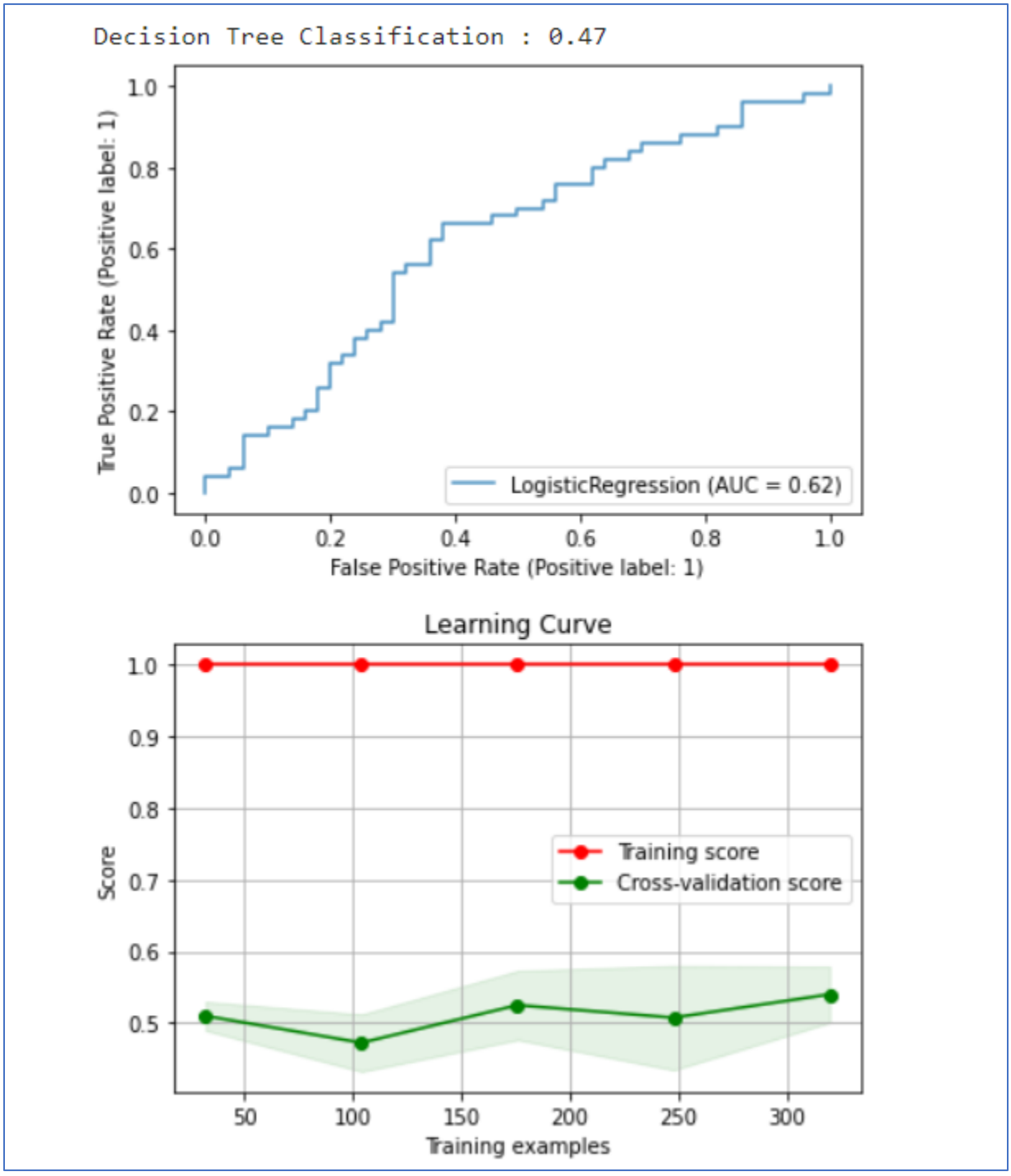

Decision Tree Classification : 0.47
True Positive Rate (Positive label: 1)
False Positive Rate (Positive label: 1)
LogisticRegression (AUC = 0.62)
Learning Curve
Score
Training examples
Training score
Cross-validation score

Image 3:

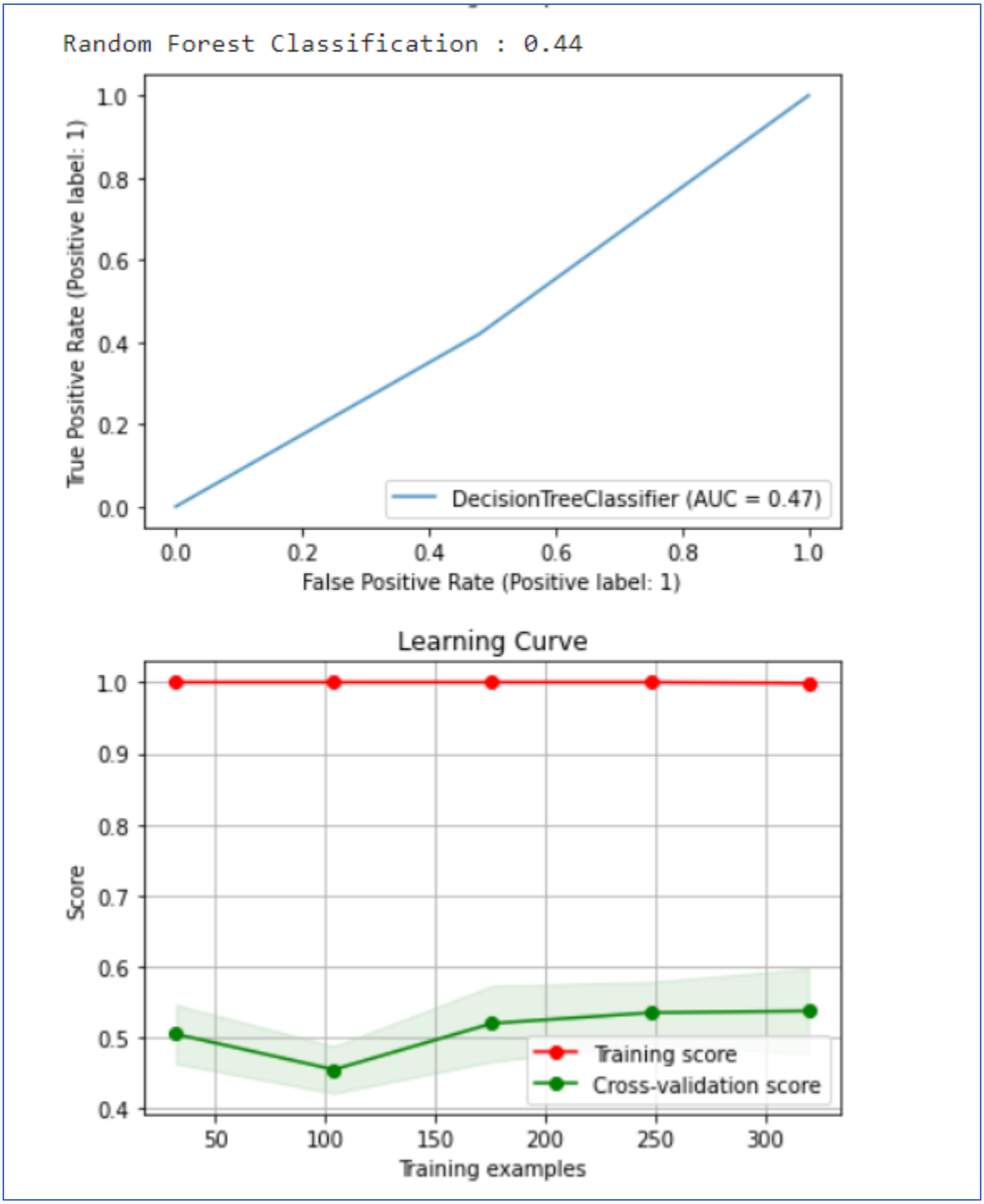

Random Forest Classification : 0.44
True Positive Rate (Positive label: 1)
False Positive Rate (Positive label: 1)
DecisionTreeClassifier (AUC = 0.47)
Learning Curve
Score
Training examples
Training score
Cross-validation score

Image 4:

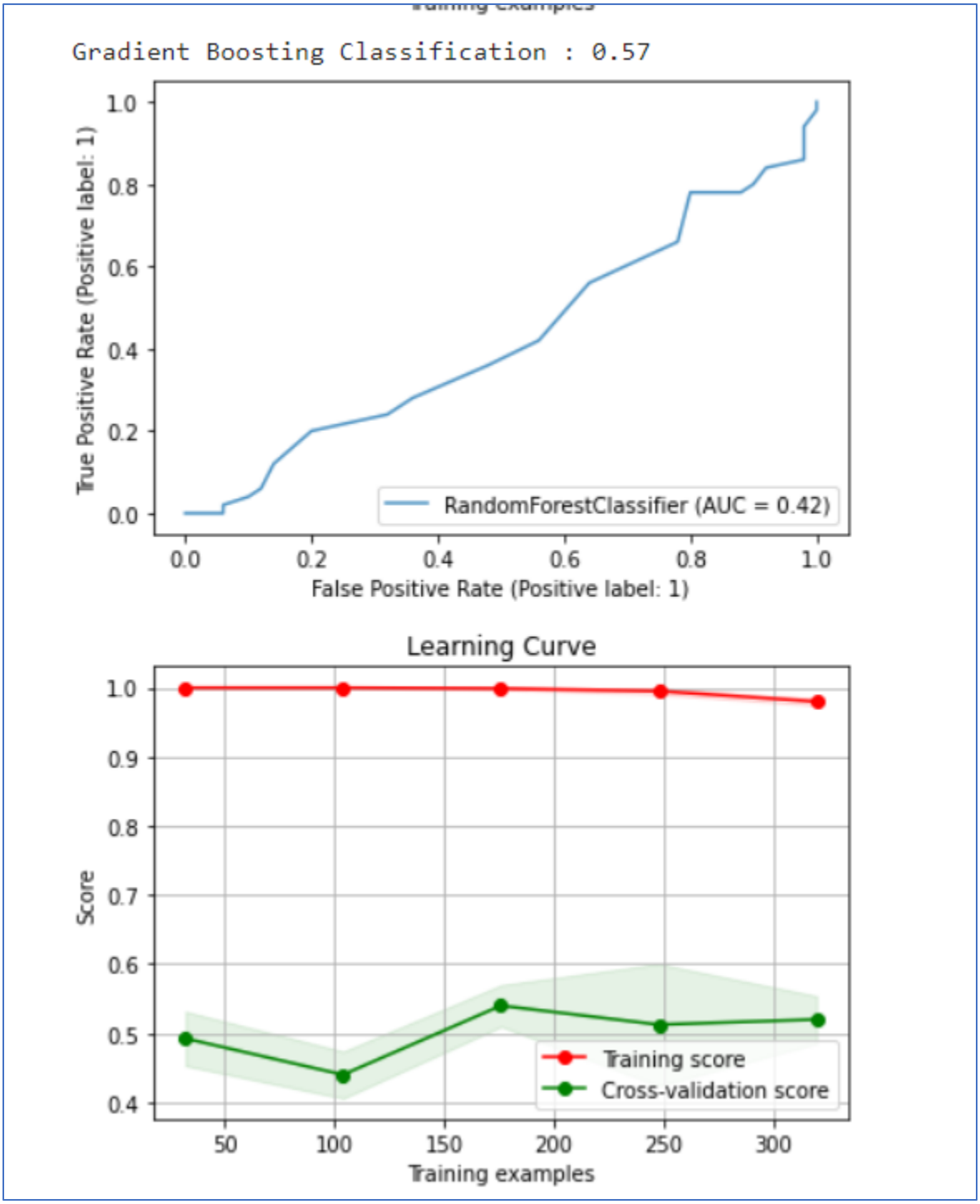

Gradient Boosting Classification : 0.57
True Positive Rate (Positive label: 1)
False Positive Rate (Positive label: 1)
RandomForestClassifier (AUC = 0.42)
Learning Curve
Score
Training examples
Training score
Cross-validation score

Image 5:

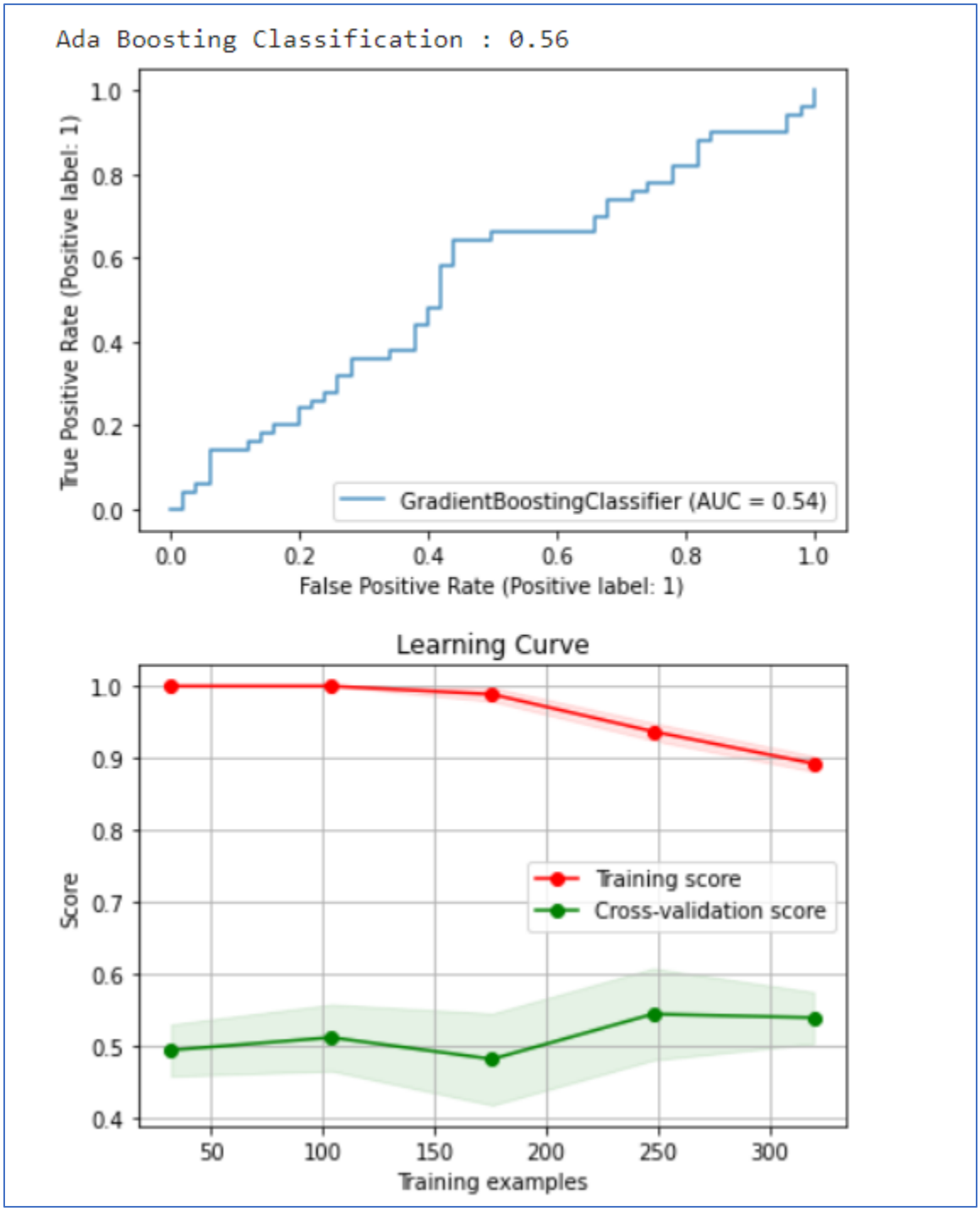

Ada Boosting Classification : 0.56
True Positive Rate (Positive label: 1)
False Positive Rate (Positive label: 1)
GradientBoostingClassifier (AUC = 0.54)
Learning Curve
Score
Training examples
Training score
Cross-validation score

Image 6:

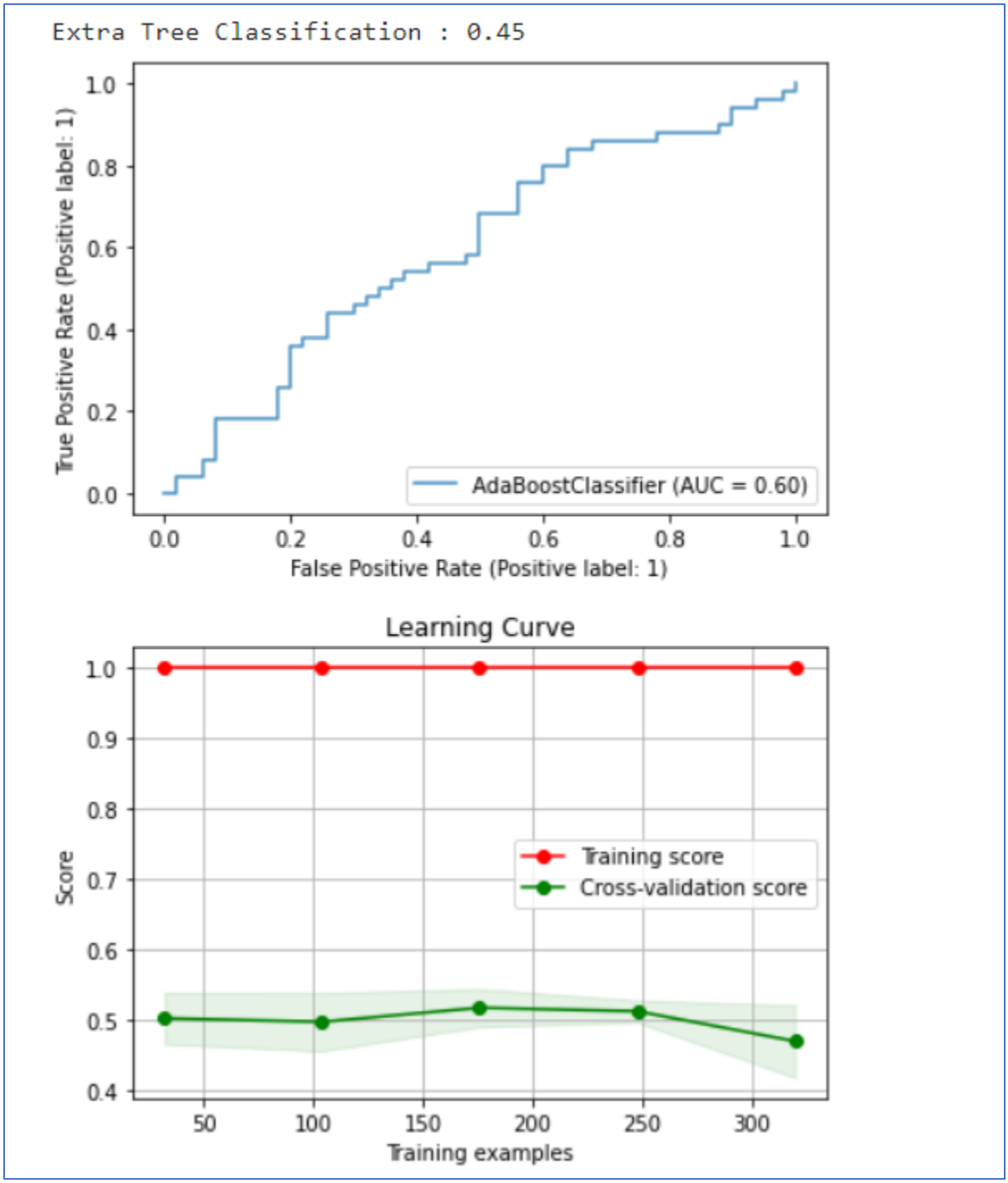

Extra Tree Classification : 0.45
True Positive Rate (Positive label: 1)
False Positive Rate (Positive label: 1)
AdaBoostClassifier (AUC = 0.60)
Learning Curve
Score
Training examples
Training score
Cross-validation score

Image 7:

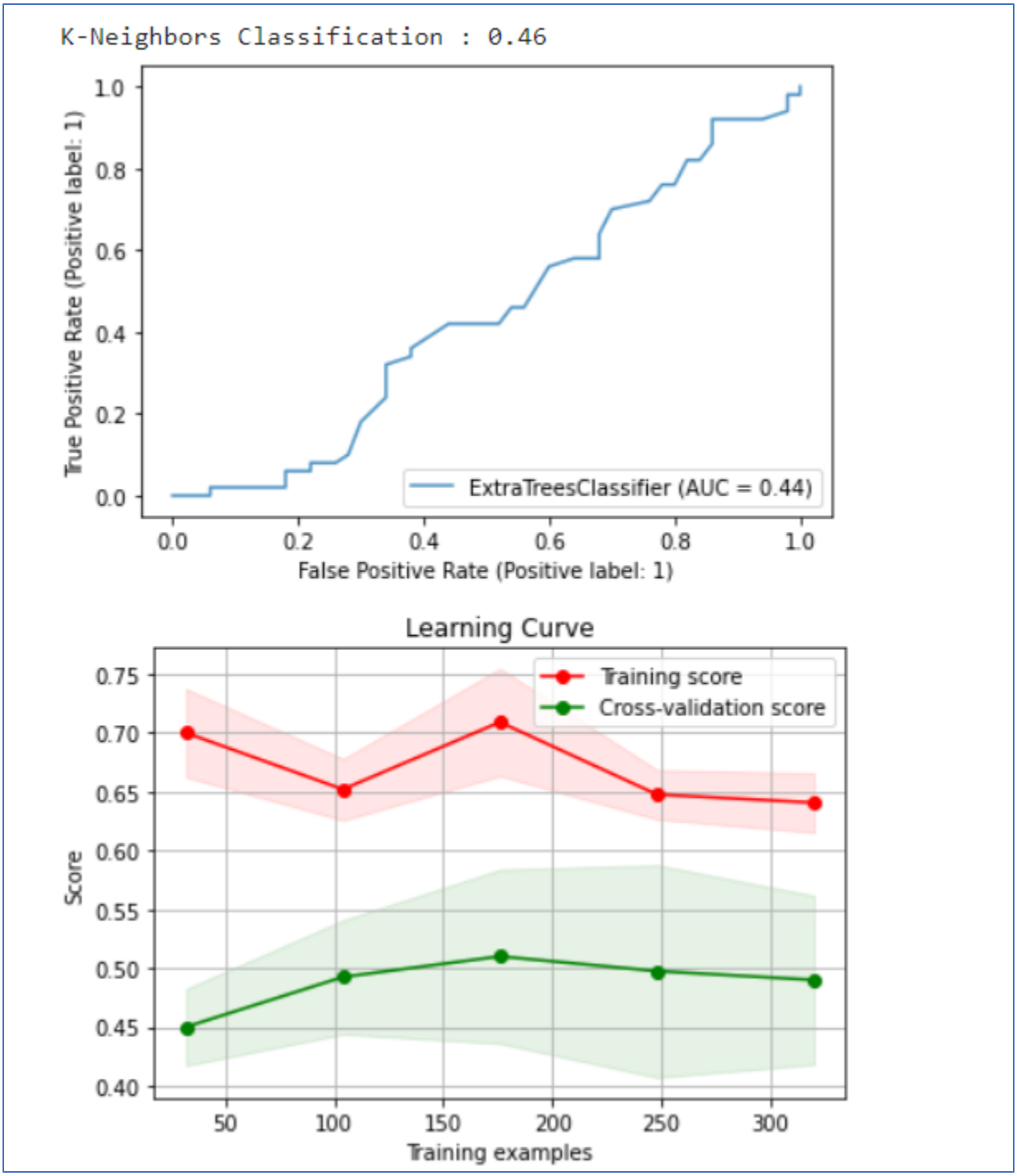

K-Neighbors Classification : 0.46
True Positive Rate (Positive label: 1)
False Positive Rate (Positive label: 1)
ExtraTreesClassifier (AUC = 0.44)
Learning Curve
Training score
Cross-validation score
Score
Training examples

Image 8:

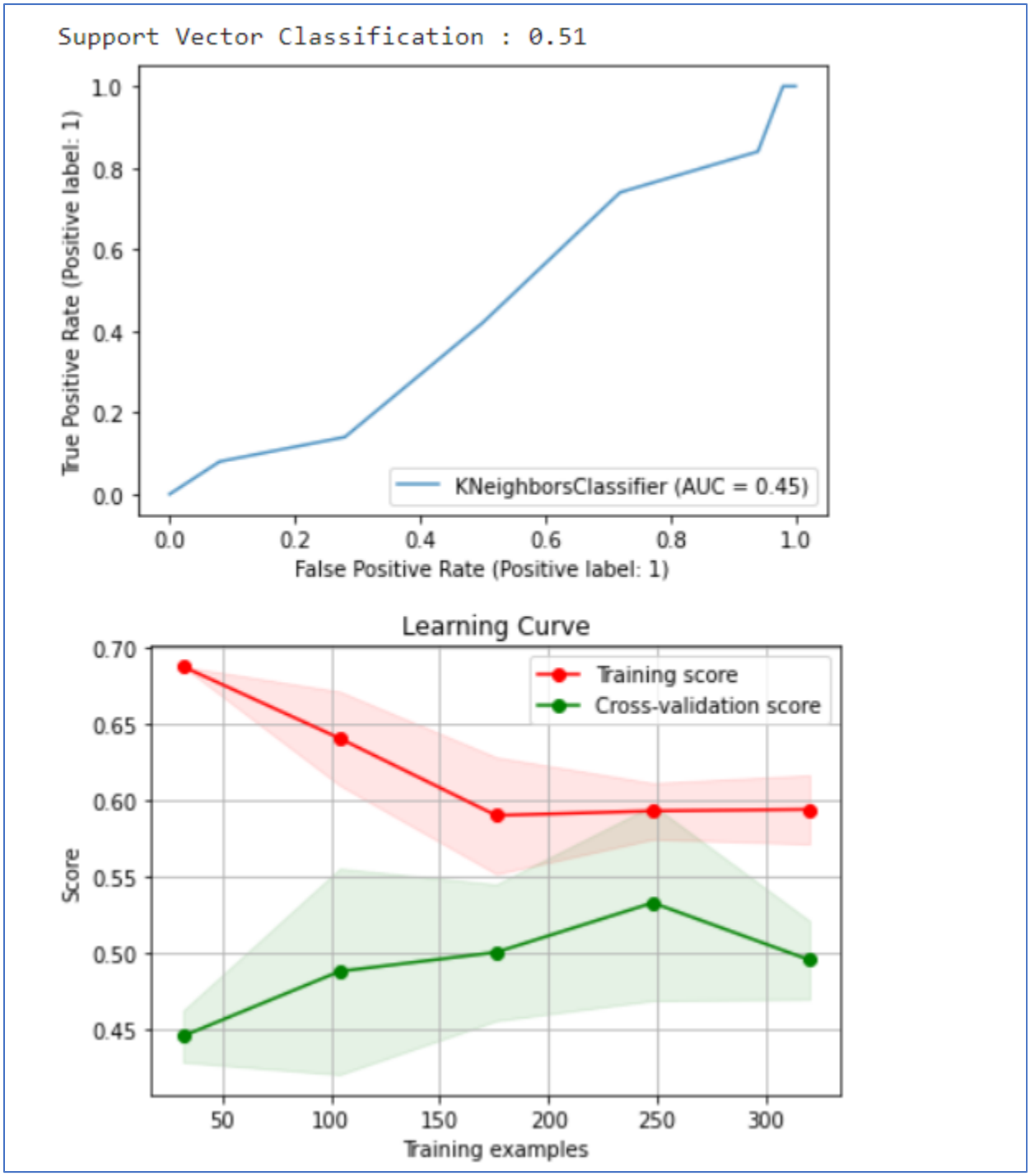
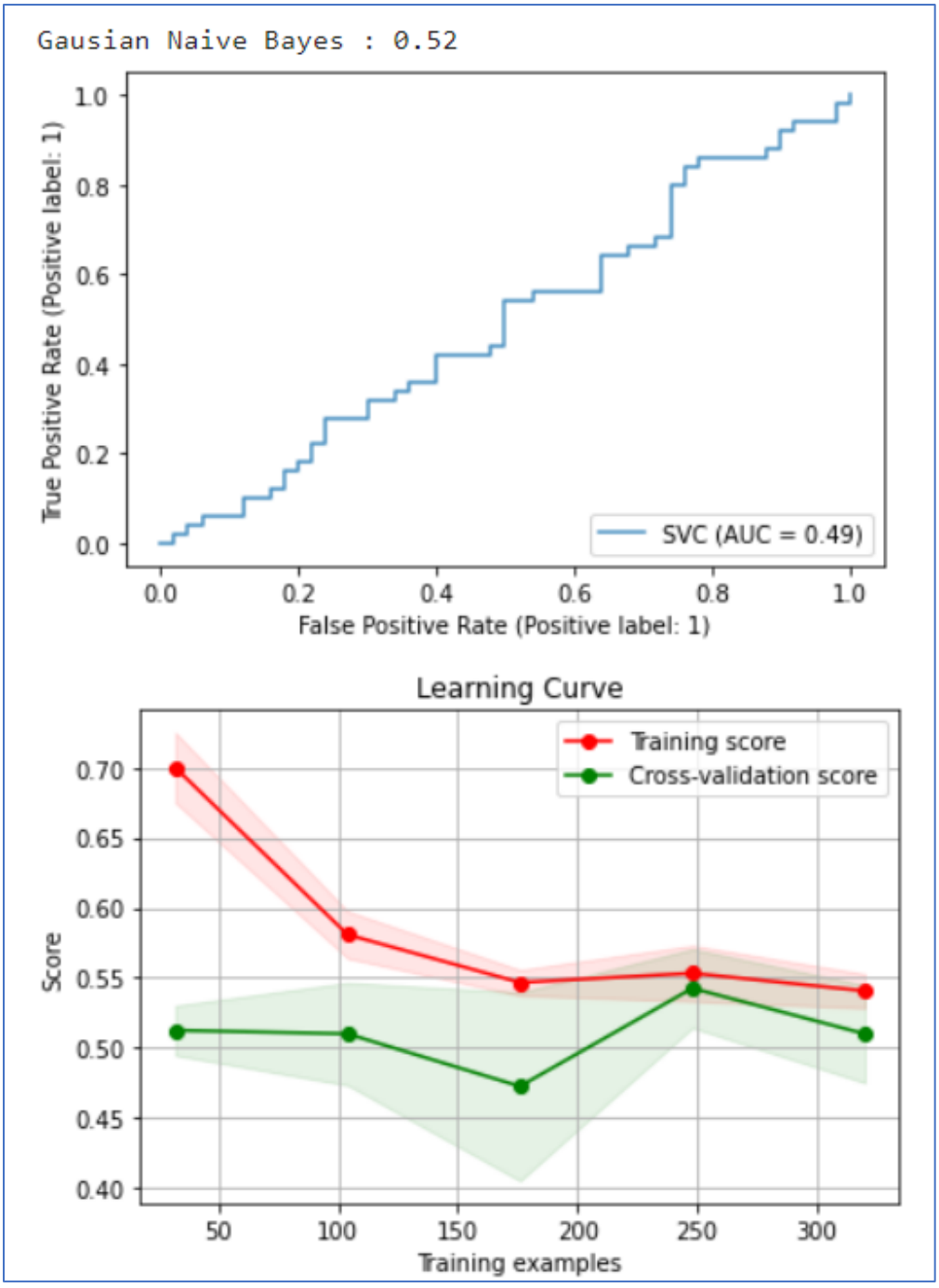

Support Vector Classification : 0.51
True Positive Rate (Positive label: 1)
False Positive Rate (Positive label: 1)
KNeighborsClassifier (AUC = 0.45)
Learning Curve
Training score
Cross-validation score
Score
Training examples

Image 9:


Gausian Naive Bayes : 0.52
True Positive Rate (Positive label: 1)
False Positive Rate (Positive label: 1)
SVC (AUC = 0.49)
Learning Curve
Training score
Cross-validation score
Score
Training examples

References

- https://towardsdatascience.com/why-1-5-in-iqr-method-of-outlier-detection-5d07fdc82097
- https://www.ibm.com/topics/logistic-regression
- https://www.saedsayad.com/decision_tree.htm#:~:text=Decision%20tree%20builds%20classification%20or,decision%20nodes%20and%20leaf%20nodes.
- https://www.analyticsvidhya.com/blog/2021/06/understanding-random-forest/
- https://monkeylearn.com/blog/introduction-to-support-vector-machines-svm/
- https://www.analyticsvidhya.com/blog/2021/09/adaboost-algorithm-a-complete-guide-for-beginners/
- https://www.geeksforgeeks.org/ml-extra-tree-classifier-for-feature-selection/
- Albashish D, Al-Sayyed R, Abdullah A, et al. Deep CNN Model based on VGG16 for Breast Cancer Classification[C]// 2021 International Conference on Information Technology (ICIT). 2021.