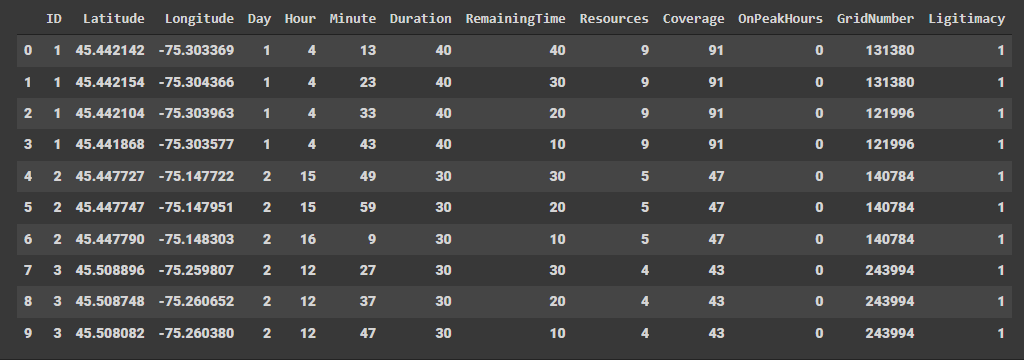
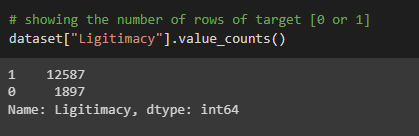
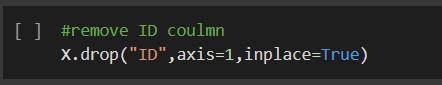
|  |  |
| --- | --- |
|  | **2022 Summer ELG 5142 Ubiquitous Sensing and Smart City Project**  Group-18 |

**1. Download MCS dataset which is used in assignment 2:**

****



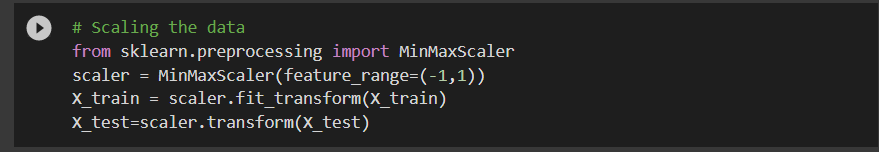
We removed useless ID column.



**2. Split the dataset into two for training (80%) and test (20%)**



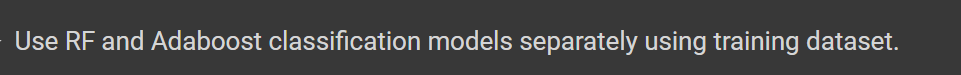
We made scaling to make all features at the same range



**3. Implement classic classifiers (Adaboost and RF)**

**4. Train Adaboost and RF via training dataset**

**5. Verify detection performance using test dataset and present results comparison in bar chart**



**Random Forest Classifier**

|  |
| --- |
|  |
|  |

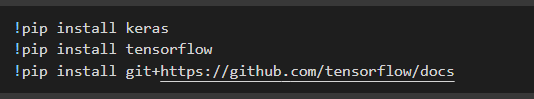
**AdaBoost Classifier**

|  |
| --- |
|  |
|  |
| Present results comparison in bar chart: |
|  |

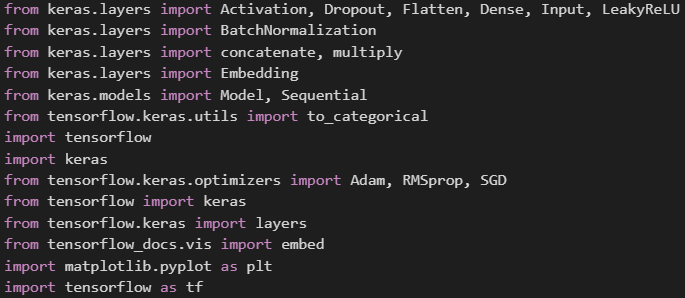
**6. Implement a CGAN model**

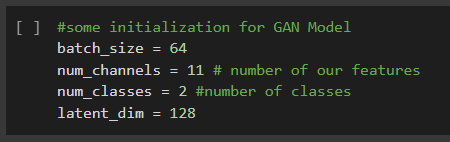
After reading the implementation manner of GAN model from [1] we made some updates on the model to make it suitable for our case.

**Install required libraries:**



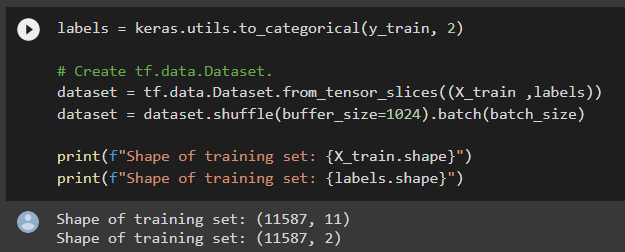
**Import required libraries:**



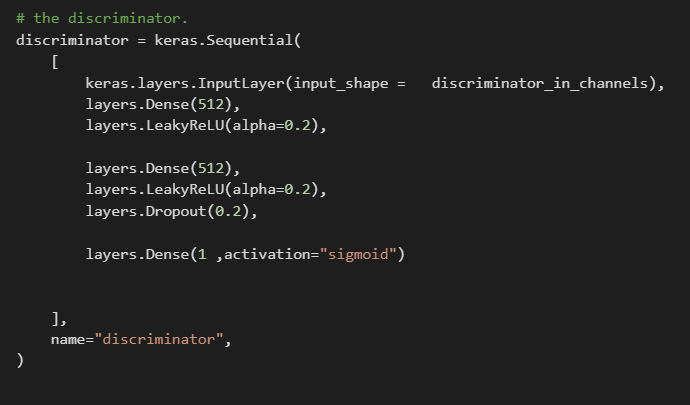


**We converted a class vector (y\_train) to binary class matrix.**

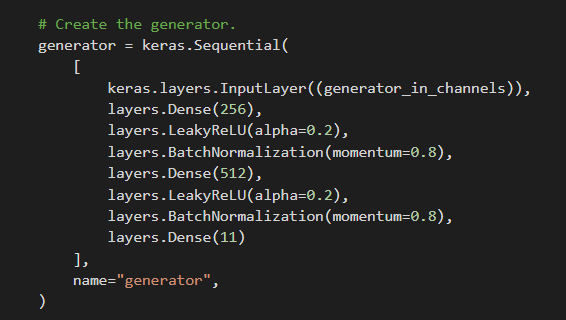
**We used tf.data.Dataset.from\_tensor\_slices () on X\_train and labels to create tf.data.Dataset (dataset).**



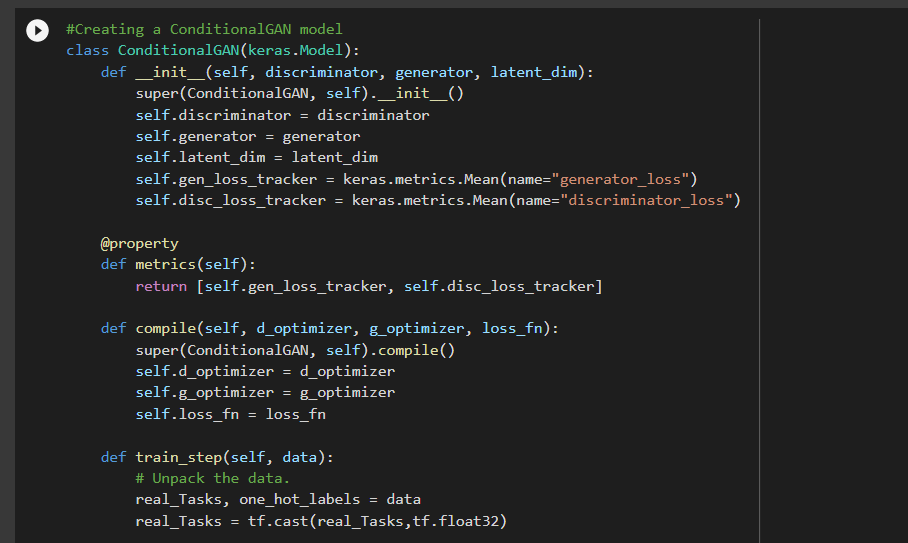
**We created the discriminator to discover the generator fault:**

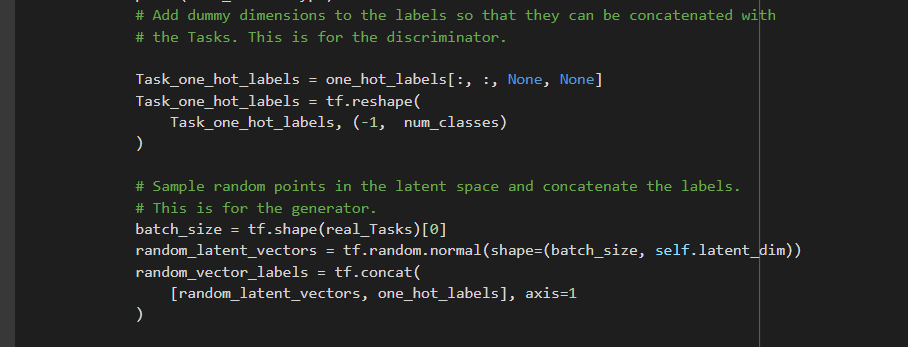


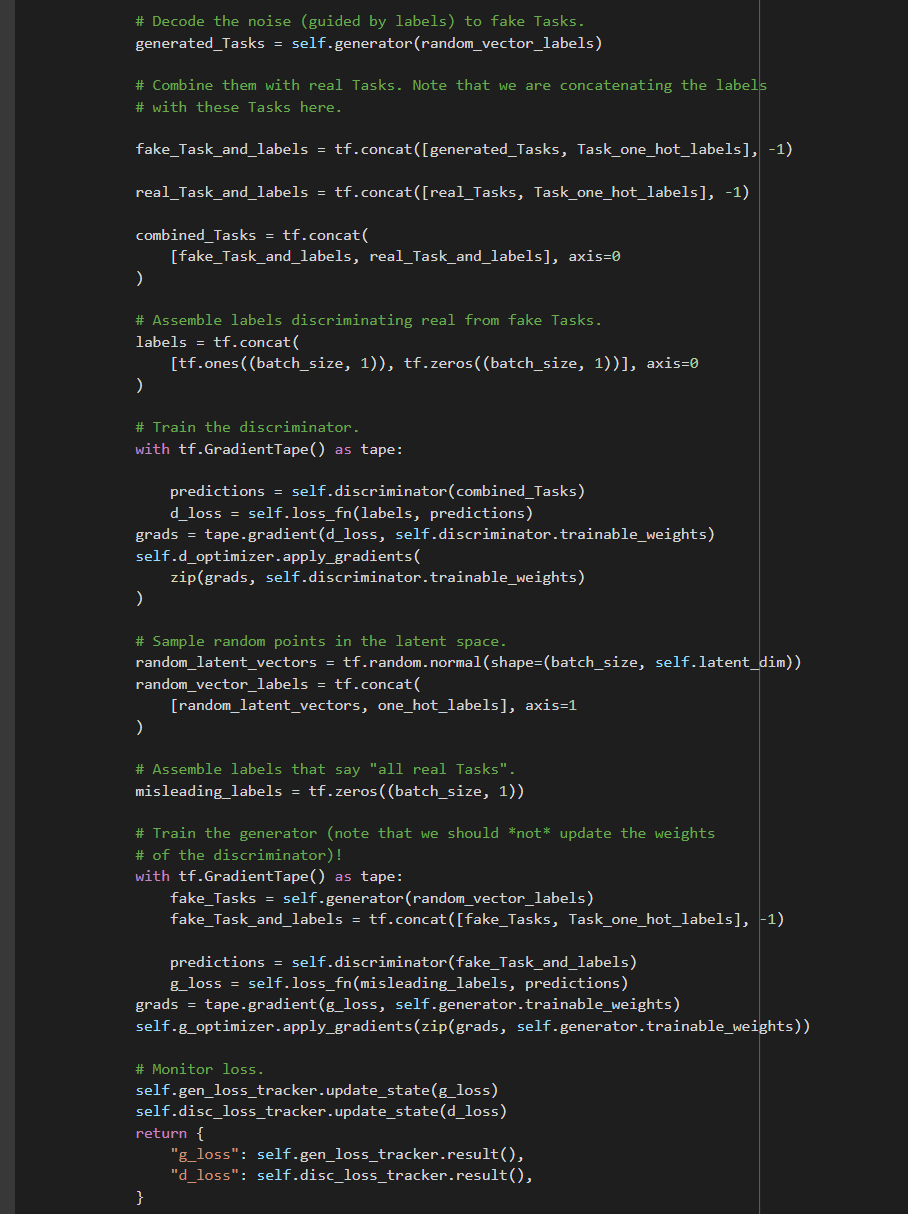
**We created the generator to generate samples:**



**We created the conditional of GAN model:**



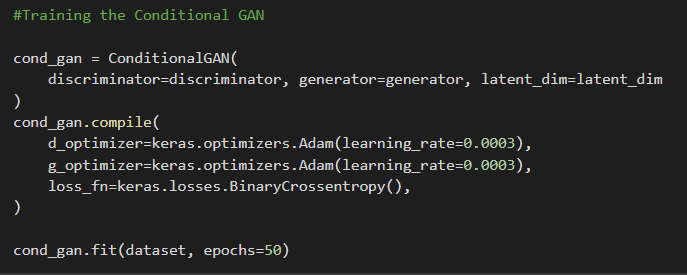




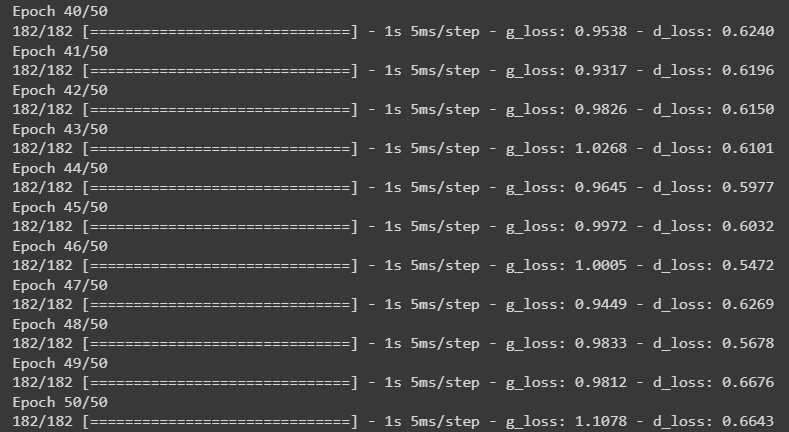
**7. Apply the provided training dataset to CGAN**

**(The training dataset can be the same as you used in assignment 2)**

**We trained the conditional GAN on 50 epochs:**

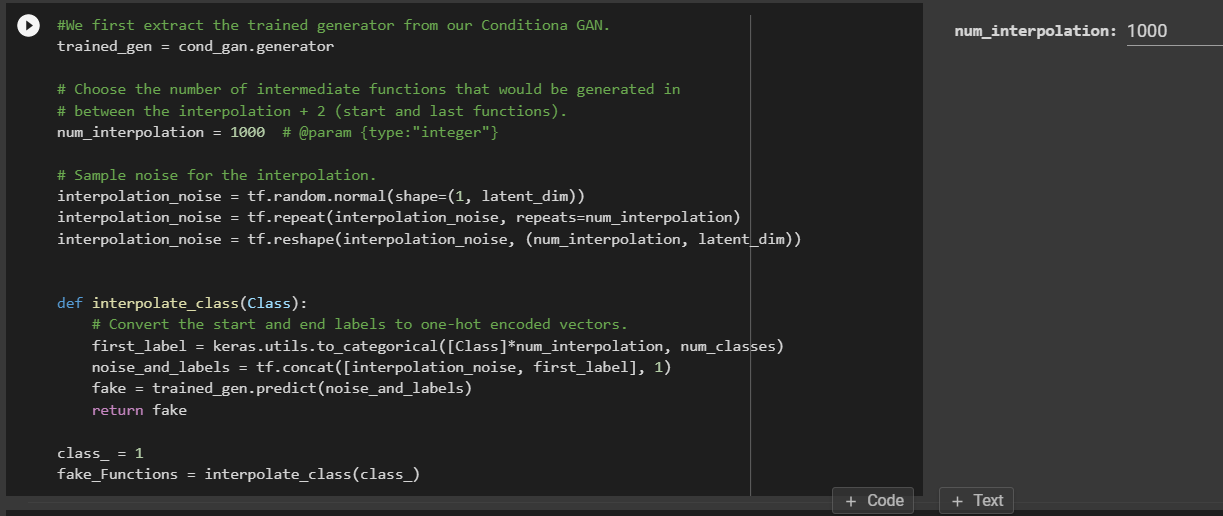


**The last 10 result of training of the conditional GAN:**

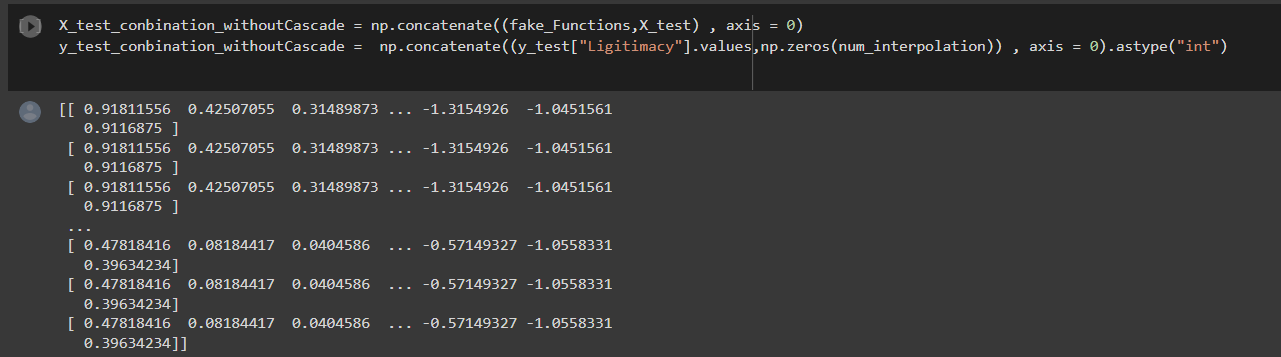


**8. Generate synthetic fake tasks via Generator network in CGAN after the training procedure**

**We generated fake 1000 samples to merge them with the original data and test our GAN model on the final combination:**



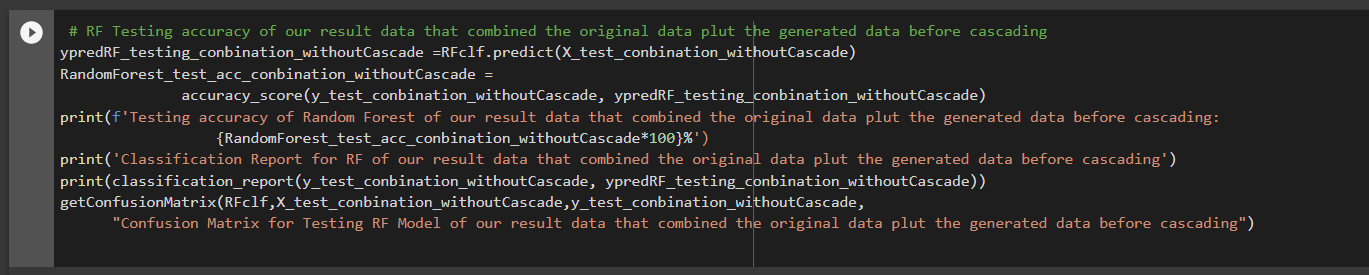
**9. Mix the generated fake tasks with the original test dataset to obtain a new test dataset.**

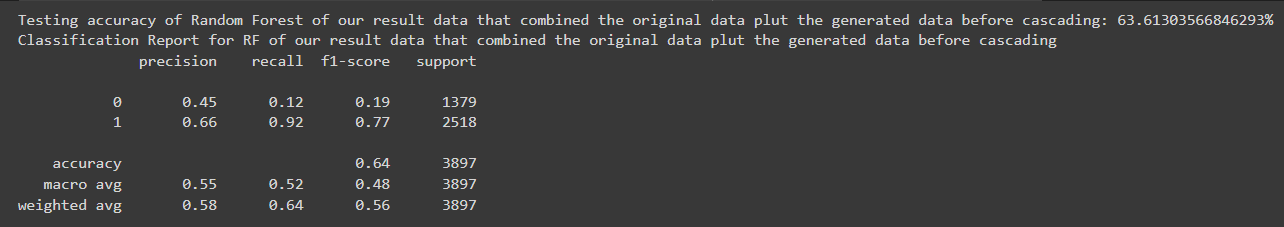


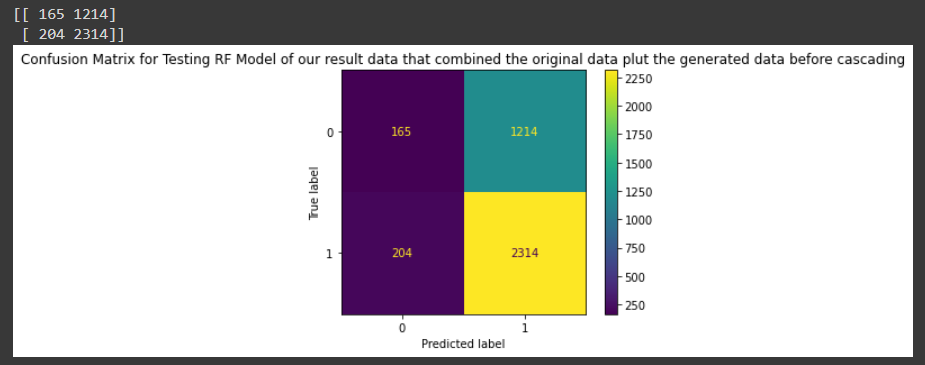
**10. Obtain Adaboost and RF detection performance using the new test dataset and present results in bar chart**

**(This step doesn’t consider Discriminator for filtering synthetic samples).**

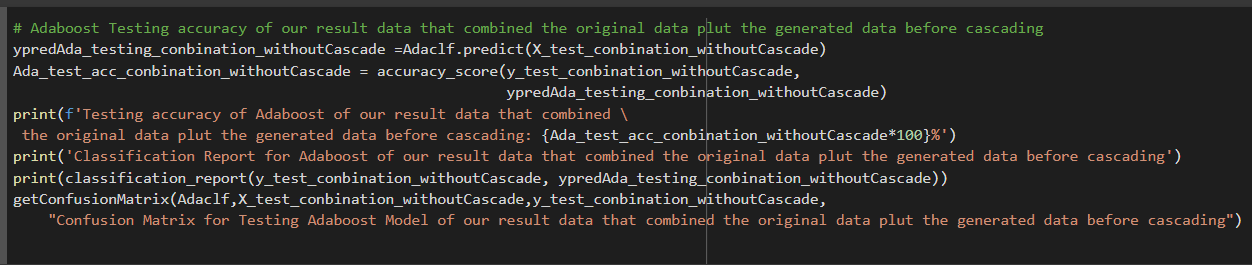
**Random Forest Classifier**

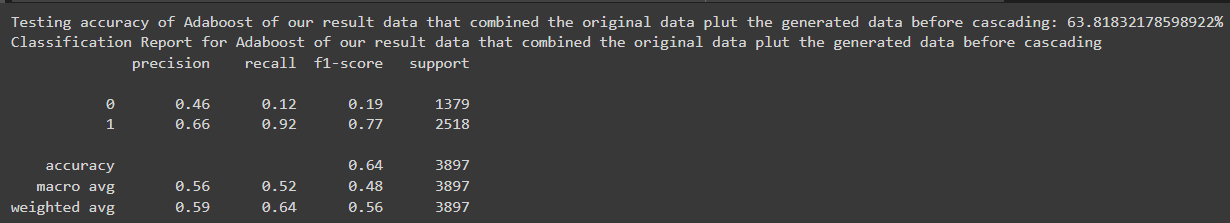


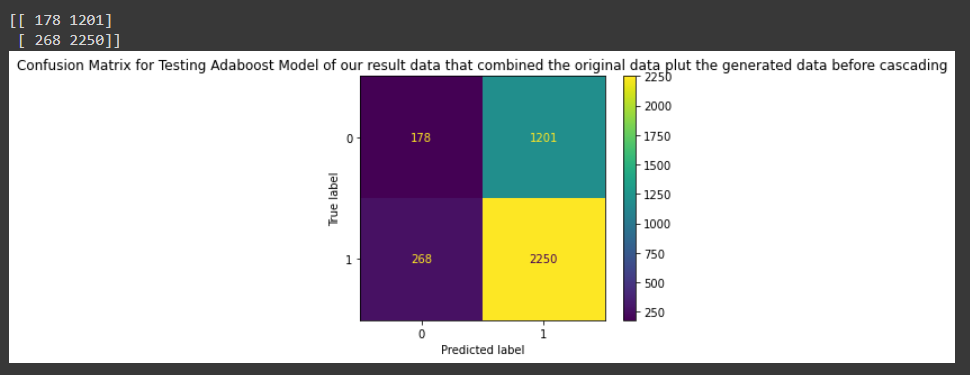




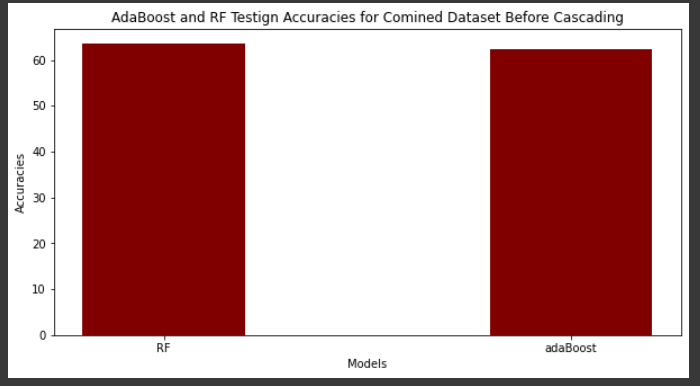
**AdaBoost Classifier**







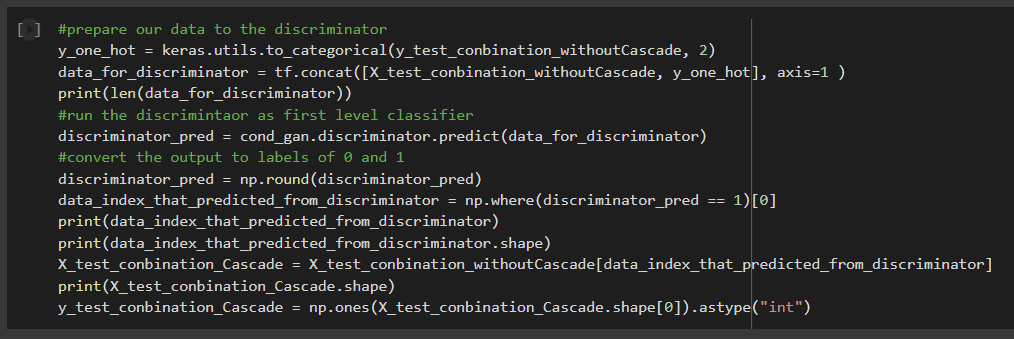
**Present results comparison in the accuracy in bar chart:**

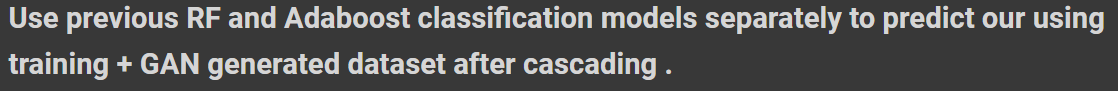
****

The above graph shows that the accuracy of two models is lower than the performance of original training data (without using GAN model).

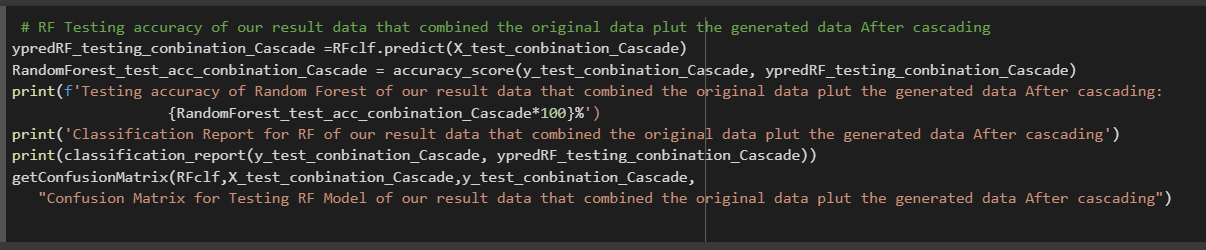
**11. According to the cascade detection framework, as shown in Figure 1, verify the cascade framework performance and show results in bar chart. Consider the Discriminator to as the first level classifier and RF/Adaboost as the second level classifier.**

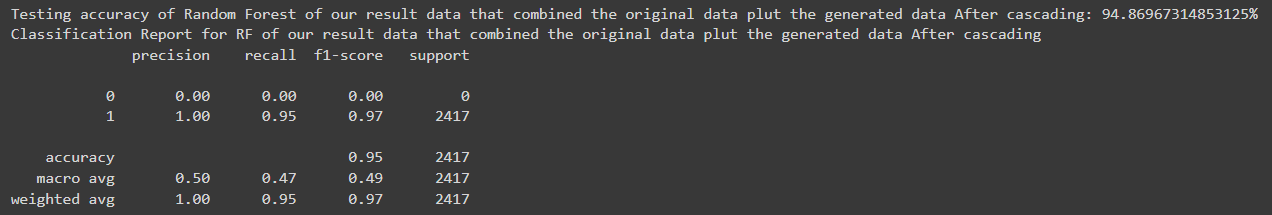
**Cascading Process to Enhance Testing Results**

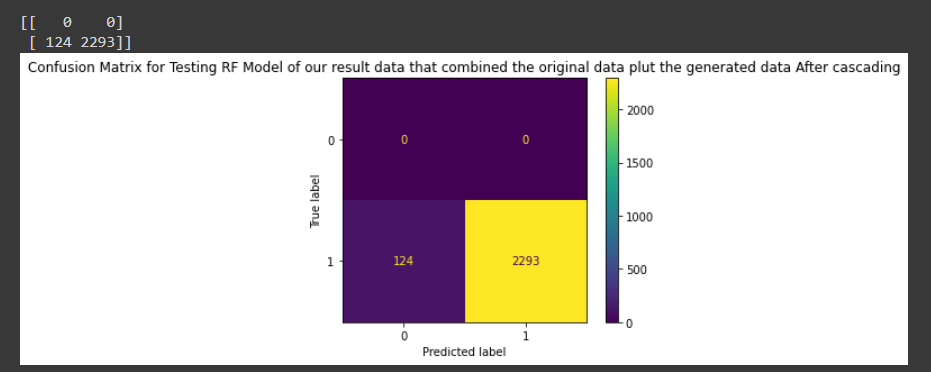




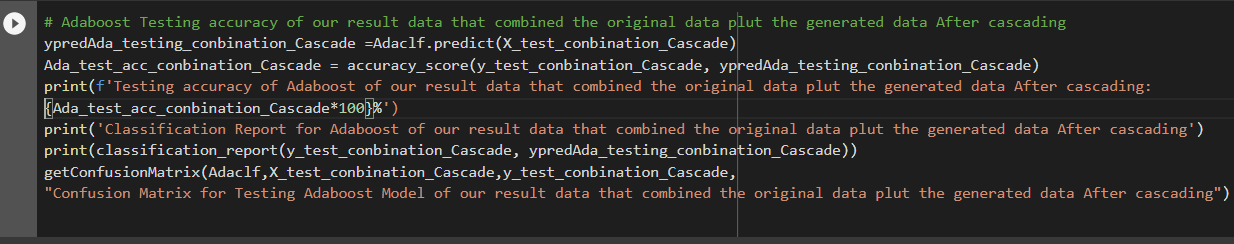
**Random Forest Classifier**

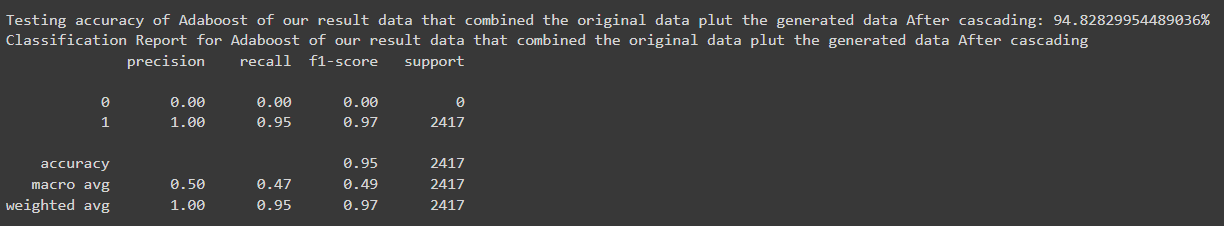






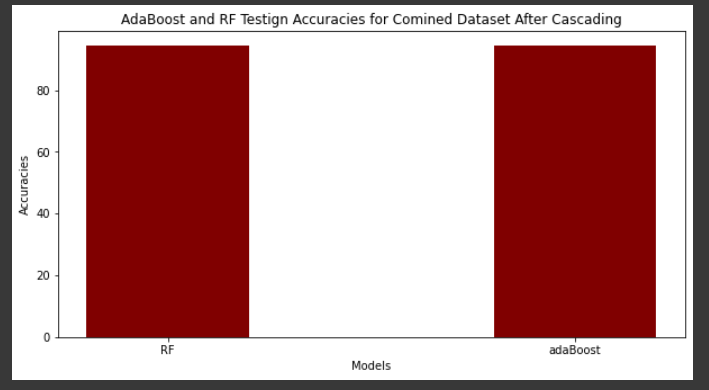
**AdaBoost Classifier**







**Present results comparison in the accuracy in bar chart:**



The above graph shows that the accuracy of two models becomes well instead of the bad performance in Question [10] because of using the discriminator (cascading).

**References:**

[1] <https://keras.io/examples/generative/conditional_gan/>