|  | ELG 20225: Smart Cities  Project  Ottawa University |
| --- | --- |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Group 15)

* Ahmed Omari
* Ali Elshenawy
* Hassan Elhefny

In this project we are going to train 2 tradition machine learning models (Random Forest and AdaBoost), to detect fake tasks. We will use an already existing dataset, that contains fake and real tasks, to train the models.

After that we are going to create a GAN neural network to generate fake tasks, then mix these tasks with the dataset we already have, and then test the two models that we have trained.

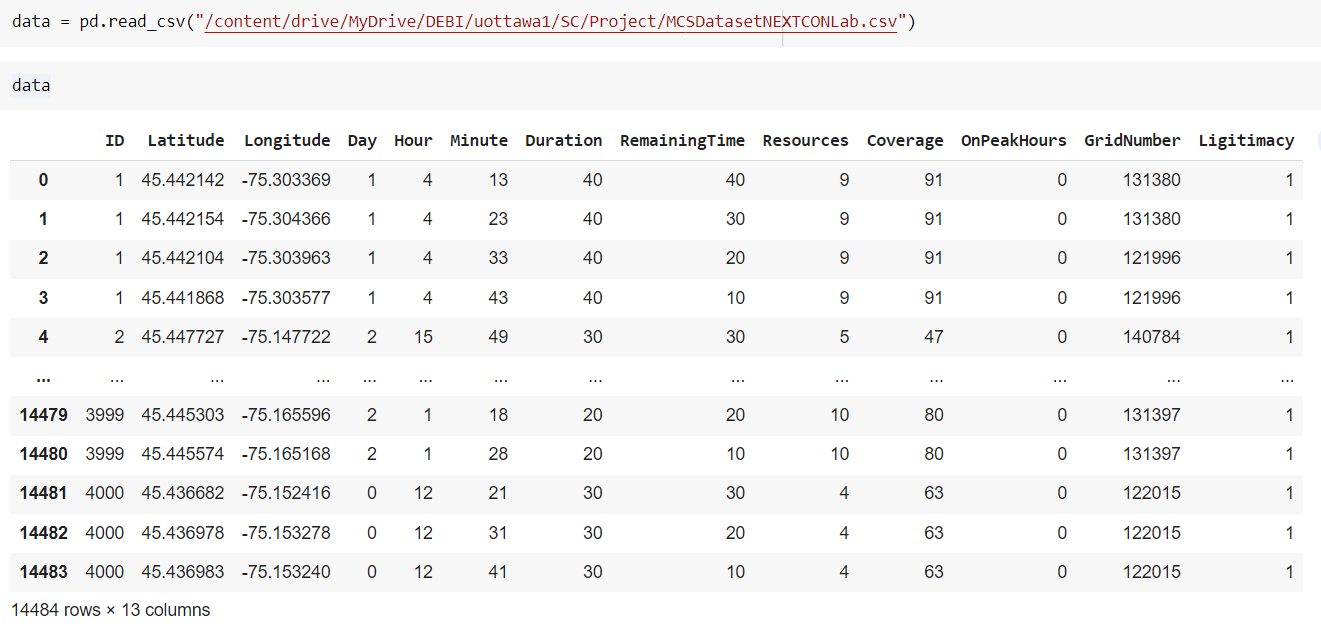
The final step is to use the discriminator of the GAN network as a first layer detector of the fake generated tasks.

And finally we will see how this process will affect the performance of our traditional machine learning models.

1. Data, and EDA:
   1. Read the data:

We read the data that was provided, for the project which is called “MCSDatasetNEXTCONLab”.

Here is the code we use to do so, and how the data looks like:

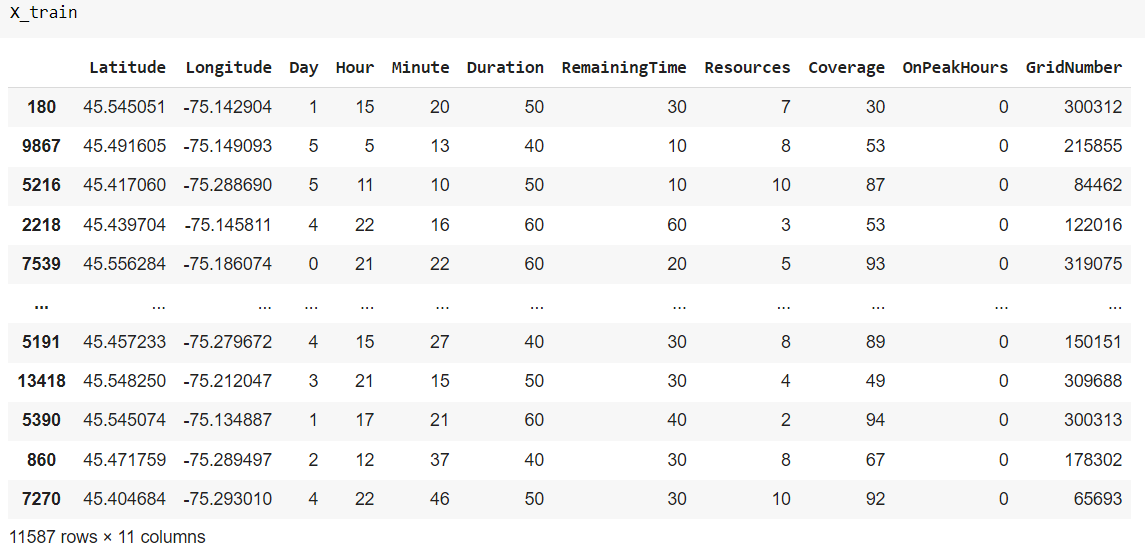


* 1. Split Data:

We split the dataset into training and testing datasets, and remove the ID column, using the following code:

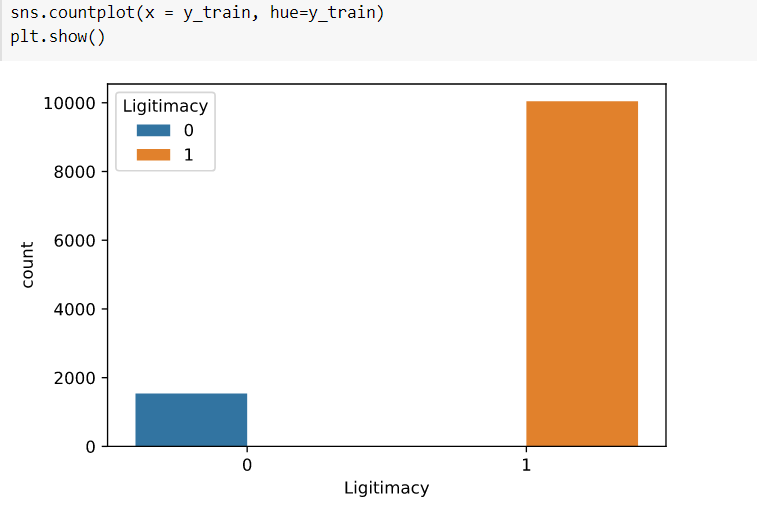


And here is how training features look like:



* 1. Check training set:

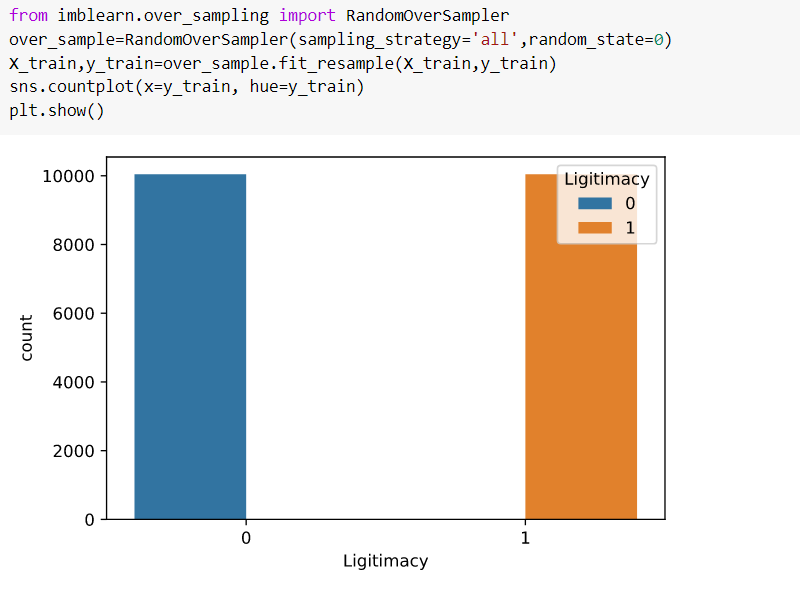
Then we check if the training dataset is not balanced.



As we can see that the training set is not balanced so the following step is using an oversampling technique to balance the data.

1. Balancing the training set:

We use OverSampling to balance the data, by the following code:

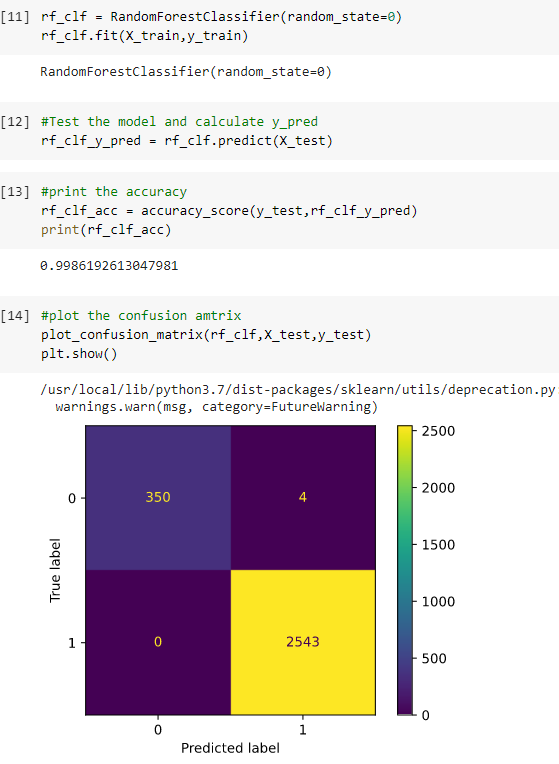


As we can see now data is balanced so let’s move to the step of building machine learning models.

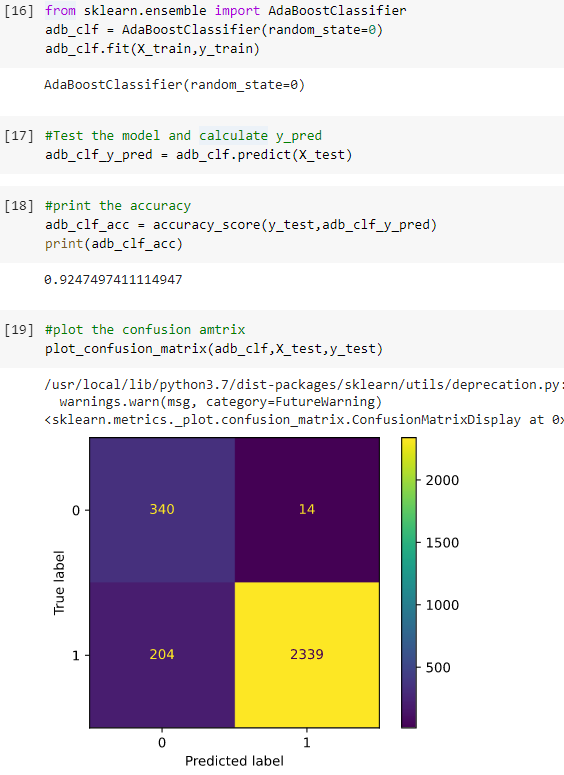
1. Building ML models:

We will build two classifiers one is random forest and the other is AdaBoos, then train them on the original test data set without any generated samples by GAN.

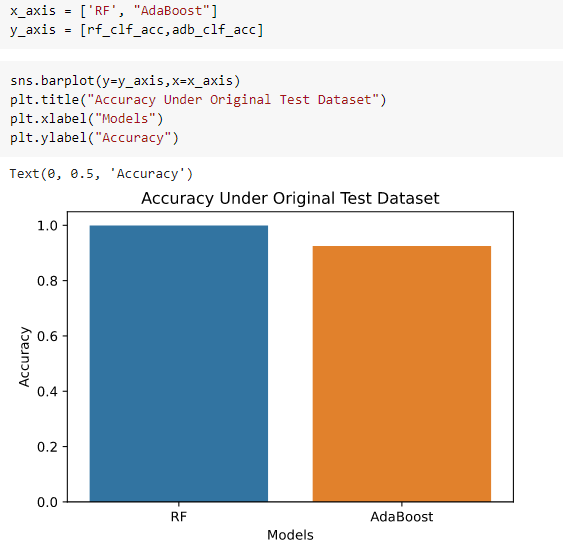
Here is the code of random forest model building, training, and evaluation:



Here is the code of AdaBoost model building, training, and evaluation:



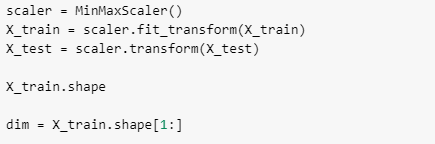
Here is a bar plot to compare between accuracies of Random Forest and AdaBoost under the original dataset:



As we can see Random Forest classifier has better performance than AdaBoost.

1. GAN:

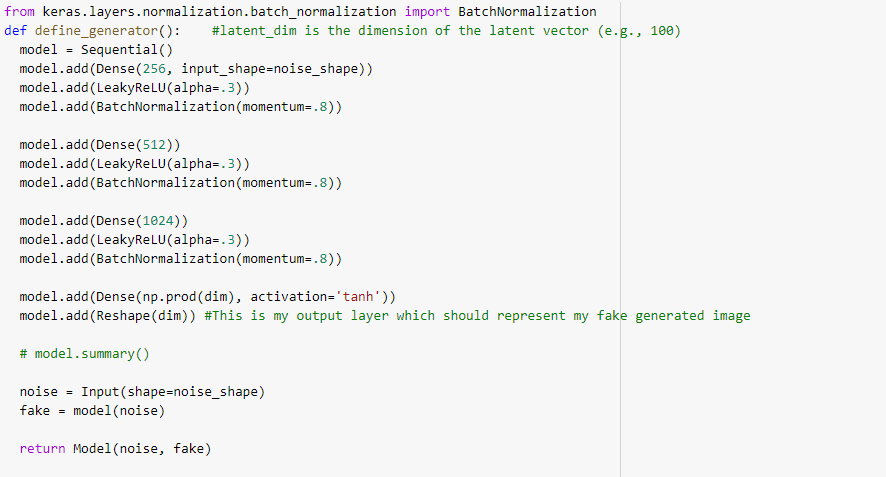
Before going the step of building the Generator or Discriminator, we first need to scale the data to be able to use it in the neural network since this gives better results.



Now we have both training and testing dataset scaled.

* 1. Build Generator:

The first step is to build the Generator model, but we should keep in mind that the output of the generator should have the same dimension shape as the data we are trying to fake. We use the following code to build the generator:



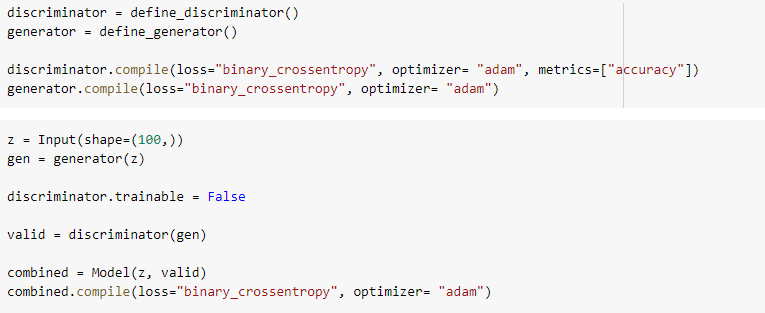
* 1. Build Discriminator:

Discriminator is nothing but a classifier whose job is to say if the data is fake or not, so it can be any model, even a traditional ML model, but we are going to build it as a neural network since this is the best model for the discriminator. Notice that the input shape of the discriminator should be the same as the output shape of the generator. We use the following code to create the discriminator:



* 1. Combine generator and discriminator:

The following step is to build the GAN model by combining the discriminator and the generator to one model using the following code:

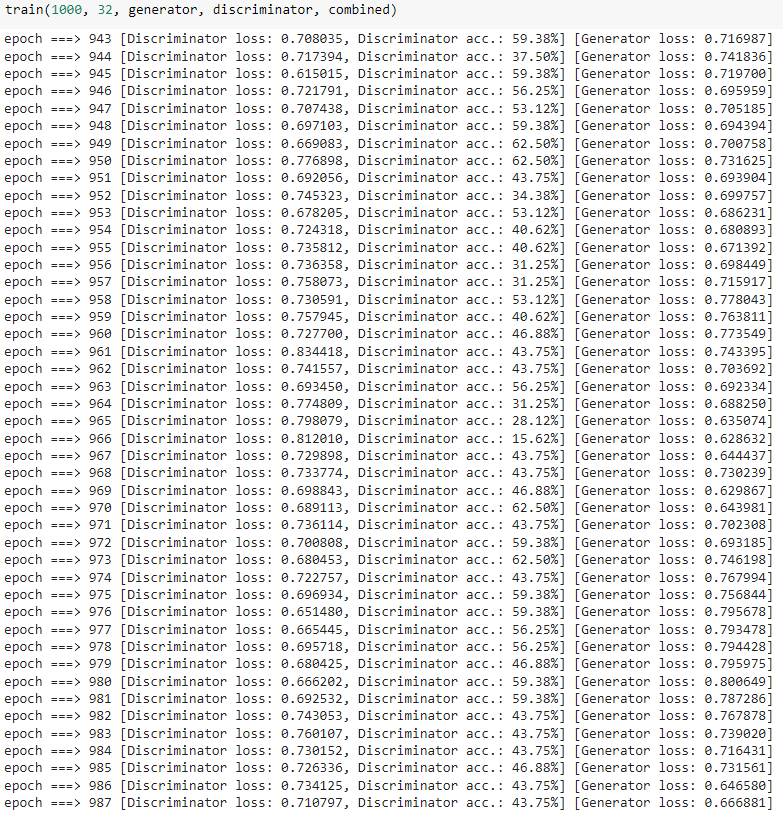


Now we have created the whole GAN model, the final step in GAN is to define the training function where the discriminator and generator will learn and fit the data, here is the code to do so:



* 1. Training:

Finally we will train the model to have it ready to generate new fake data:

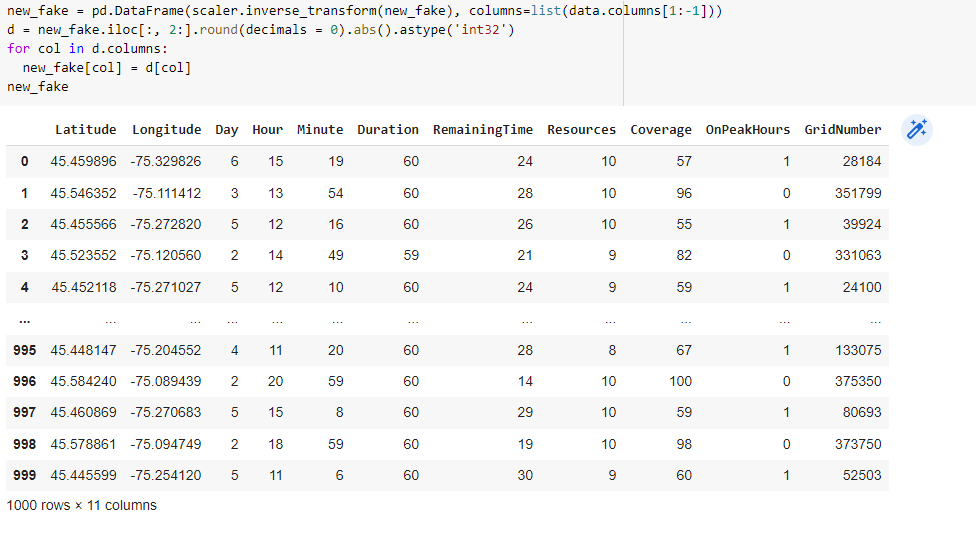


1. Generate Fake Data:

Now we will generate fake data using the generator we have trained, as following:

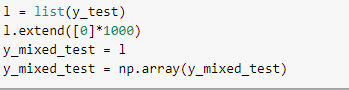


Now here is our generated data, but as we can see the data is scaled, and this is normal since we scaled the training data before we pass it into GAN model, so we need to apply reverse transforming using the following code:

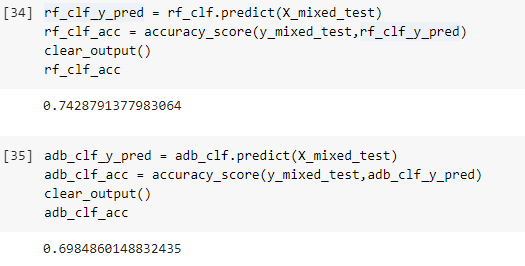


Now we have the generated data that looks like the real one. The next step is to mix it with the test data that we already have.

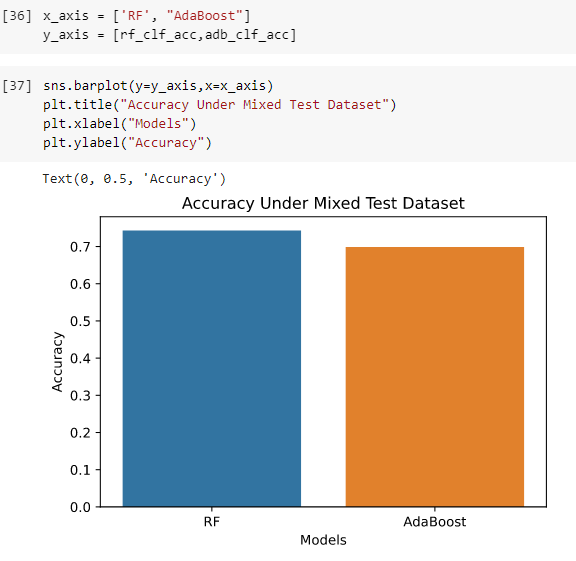




1. Test Random Forest and AdaBoost models on Mixed data:



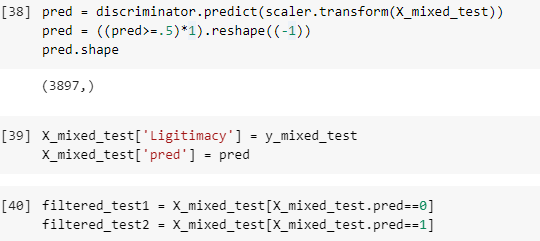
Here is a comparison between their accuracies:



As we can see, the Random Forest model has better performance.

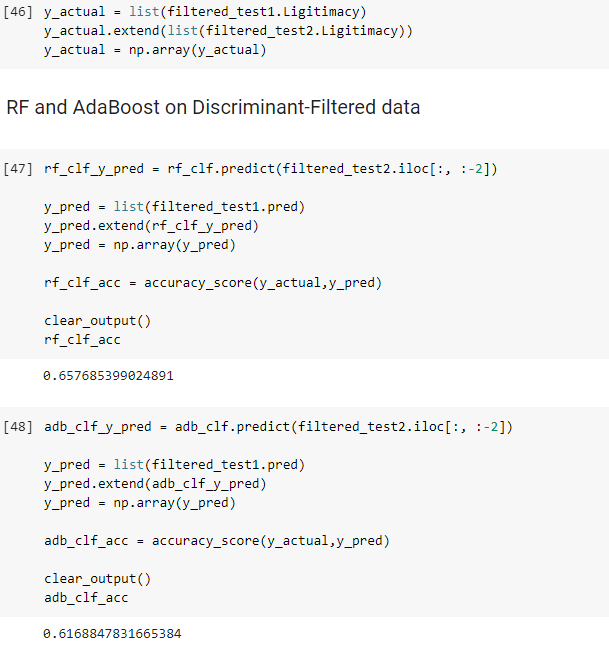
1. Cascaded FrameWork:

At first we will use the discriminator a first filtering layer as following:

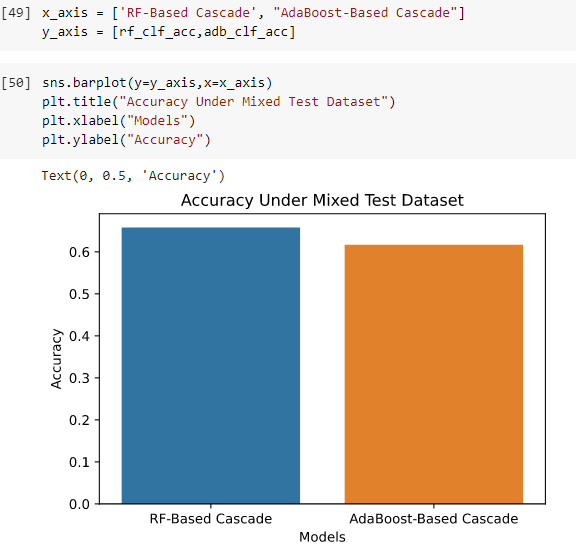


Now the samples that were classified to be fake will not be passed to the second layer, and only the samples that were classified as real will be passed to be tested again on Random Forest and AdaBoost models.

Here is the code of the second layer:



Here is a bar plot that shows comparison between classifiers of the cascade framework:



As we can see here Random forest has better performance, as well.

1. Conclusion:
   1. ML models (RF&Ada) did a good job in identifying the fake tasks in the original dataset where both had accuracy values near to 100%.
   2. After we generated fake data and mixed it with the test dataset the performance of the two models went down which means the GAN generated data was close to the real data.
   3. When we put the discriminator as a first layer classifier this caused the performance of the two models to go down, which means that the discriminator classified some samples wrongly.
   4. So we can say that GAN has a strong generator but the discriminator is not as good as expected to be.
   5. Finally Random Forest classifier has better performance in classifying the provided data.