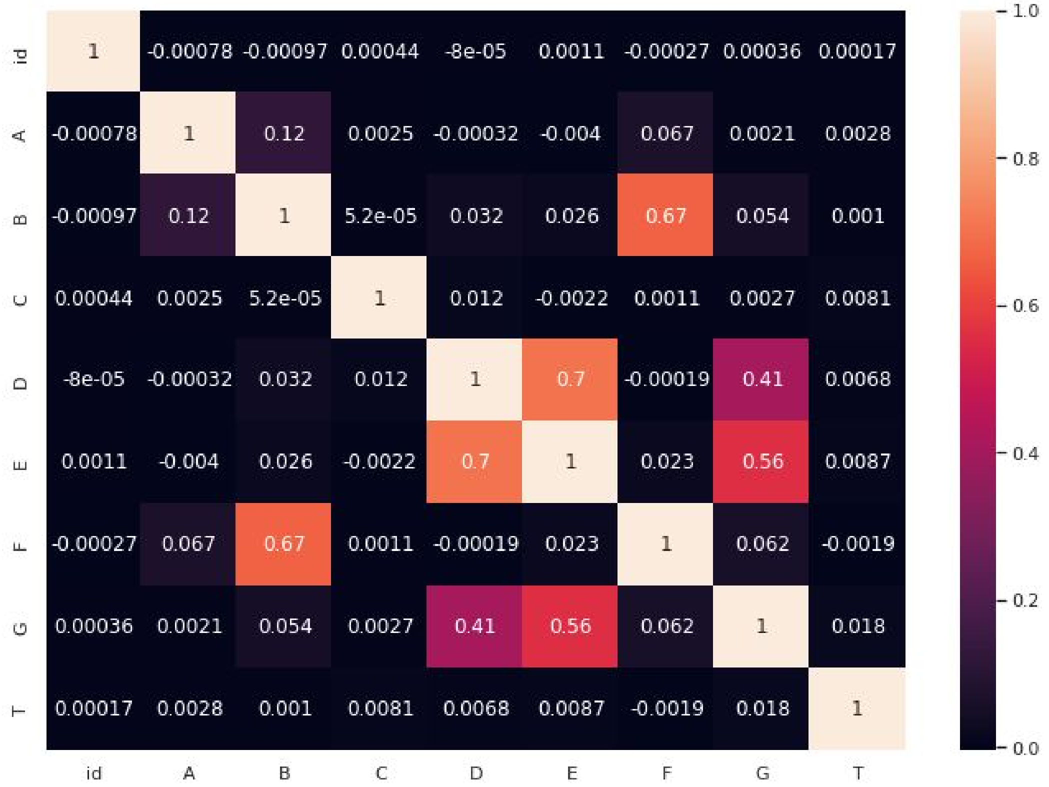
# Data Mining
# Assignment-2

**Submitted by:**
**Nitindeep Singh (MT19069)**
**Gaurav Lodhi (MT19063)**

## Preprocessing:

We imported the given_dataset.csv file to a data frame using the pandas's library. We removed the id column and separated all the class labels from this data frame. Now we performed the following steps:

1. We plotted a heatmap to plot the correlation between the features as shown in the following figure.

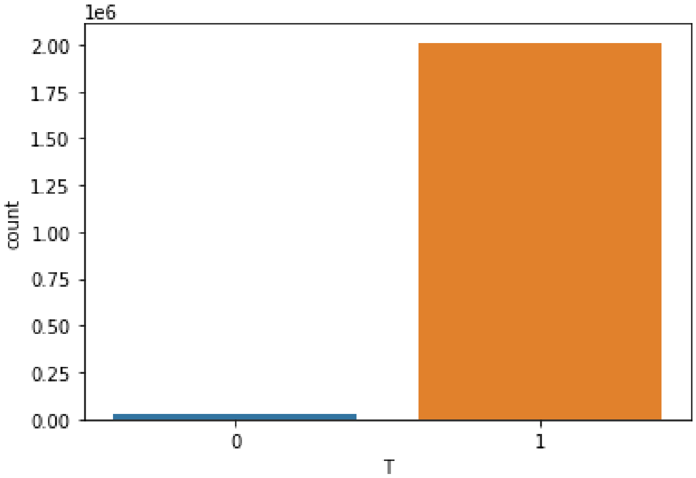


| | id | A | B | C | D | E | F | G | T |
|---|---|---|---|---|---|---|---|---|---|
| id | 1 | -0.00078 | -0.00097 | 0.00044 | -8e-05 | 0.0011 | -0.00027 | 0.00036 | 0.00017 |
| A | -0.00078 | 1 | 0.12 | 0.0025 | -0.00032 | -0.004 | 0.067 | 0.0021 | 0.0028 |
| B | -0.00097 | 0.12 | 1 | 5.2e-05 | 0.032 | 0.026 | 0.67 | 0.054 | 0.001 |
| C | 0.00044 | 0.0025 | 5.2e-05 | 1 | 0.012 | -0.0022 | 0.0011 | 0.0027 | 0.0081 |
| D | -8e-05 | -0.00032 | 0.032 | 0.012 | 1 | 0.7 | -0.00019 | 0.41 | 0.0068 |
| E | 0.0011 | -0.004 | 0.026 | -0.0022 | 0.7 | 1 | 0.023 | 0.56 | 0.0087 |
| F | -0.00027 | 0.067 | 0.67 | 0.0011 | -0.00019 | 0.023 | 1 | 0.062 | -0.0019 |
| G | 0.00036 | 0.0021 | 0.054 | 0.0027 | 0.41 | 0.56 | 0.062 | 1 | 0.018 |
| T | 0.00017 | 0.0028 | 0.001 | 0.0081 | 0.0068 | 0.0087 | -0.0019 | 0.018 | 1 |

From this heatmap, we found out that there aren't significant correlations between two attributes so all the features can be used for further analysis.

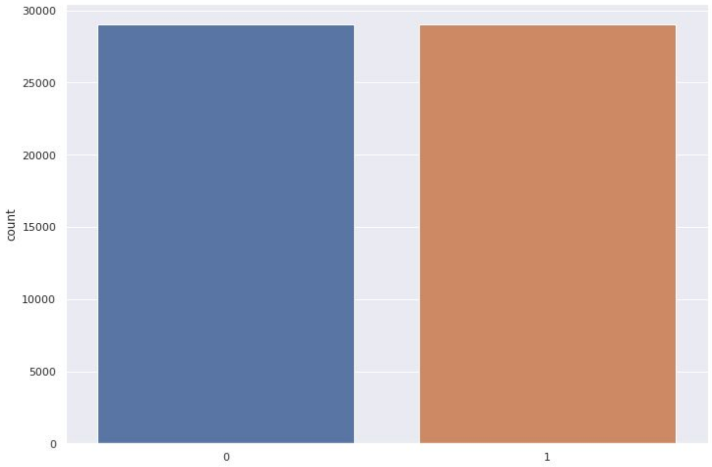
2. We plot the count plot to see the class label distribution and we found that the given dataset is an imbalanced dataset with the following distribution.

   ```
   Class 1: 2012677,
   Class 0: 29010
   ```



3. To overcome we used different sampling on the given dataset as follows.
   a. Random Under Sampling:- To make the dataset balanced we used imblearn's RandomUnderSampling and reduced all the class 1 data points to equal to no of samples of class 0 as 29010.
   b. Random over Sampling:- To make the dataset balanced we used imblearn's RandomOverSampling classifier and oversample class 0 and make it equal to the no of class 1 samples. But by Random Over Sampling, we couldn't get significant results so we dropped the idea to use it.

After performing the sampling techniques we found the following class distribution in the preprocessed dataset.



4. We split the dataset for the train (90%) and validation (10%).

## Methodology:

We analyzed the performance of many different machine learning models given the following.

1. **KNN:-** We apply K(8)- nearest neighbor algorithm on the preprocessed dataset on different parameters but we get a maximum Auc/Roc score of **0.7478205414451521**.

2. **Decision Tree-Based ensemble methods** as following:
    a. **Decision tree:** We applied a simple Decision Tree algorithm on the preprocessed dataset and get the highest AucRoc score as **0.7045178574377979.**

    b. **RandomForestClassifier: 0.7798918264674858**
    RandomForestClassifier(n_jobs=-1,n_estimators=150,max_depth=35,max_features=4,random_state=0)

    c. **XGBoost:** I used Tree-based Ensemble model XGBoost and found some significant amount of improvement in the Auc/Roc score so i tried variation in the hyperparameters and found the following observations.
    These observations are based on probability scores:-

    (objective="binary:logistic",learning_rate =0.01, n_estimators=1000, max_depth=12, min_child_weight=1, gamma=0, subsample=0.8, colsample_bytree=0.8, nthread=4, scale_pos_weight=1, seed=27)
    **training = 0.88588955429572**
    **testing = 0.88895**

    (objective="binary:logistic",learning_rate =0.01, n_estimators=1200, max_depth=12, min_child_weight=1, gamma=0, subsample=0.8, colsample_bytree=0.8, nthread=4, scale_pos_weight=1, seed=27)
    **training = 0.8863458423189793**
    **testing = 0.88950**

    (objective="binary:logistic",learning_rate =0.01, n_estimators=1400, max_depth=12, min_child_weight=1, gamma=0, subsample=0.8, colsample_bytree=0.8, nthread=4, scale_pos_weight=1, seed=27)
    **training = 0.8863458423189793**
    **testing = 0.88950**

(objective="binary:logistic",learning_rate =0.01, n_estimators=1500, max_depth=12, min_child_weight=1, gamma=0, subsample=0.8, colsample_bytree=0.8,  nthread=4, scale_pos_weight=1, seed=27)
**training = 0.8867718893781971**
**testing =  0.88950**

(objective="binary:logistic",learning_rate =0.01, n_estimators=2700, max_depth=12, min_child_weight=1, gamma=0, subsample=0.8, colsample_bytree=0.8,  nthread=4, scale_pos_weight=1, seed=27)
**training = 0.887034195579068**
**testing =  0.88997**

**Observations based on the discrete class labels**: I tried discrete class submission on all the above submissions but found the following as the best Auc/Roc score:

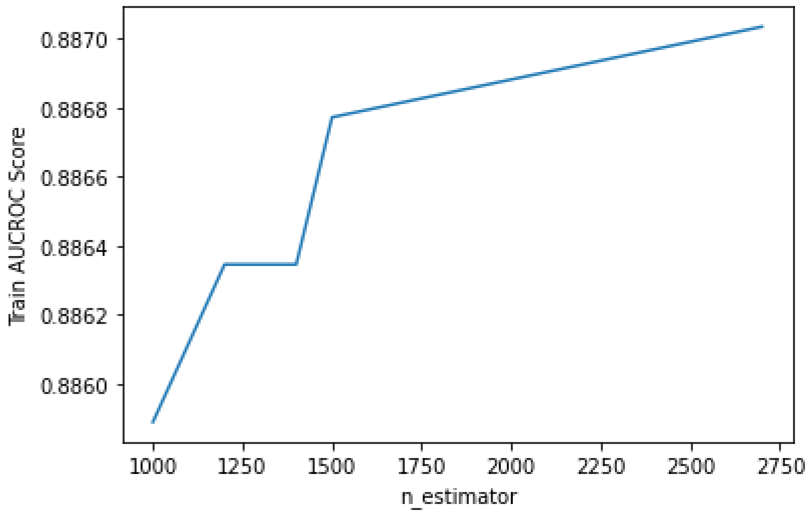
(objective="binary:logistic",learning_rate =0.01, n_estimators=1400, max_depth=14, min_child_weight=1, gamma=0, subsample=0.8, colsample_bytree=0.8,  nthread=4, scale_pos_weight=1, seed=27)
**training = 0.80170**
**testing =  0.80033**

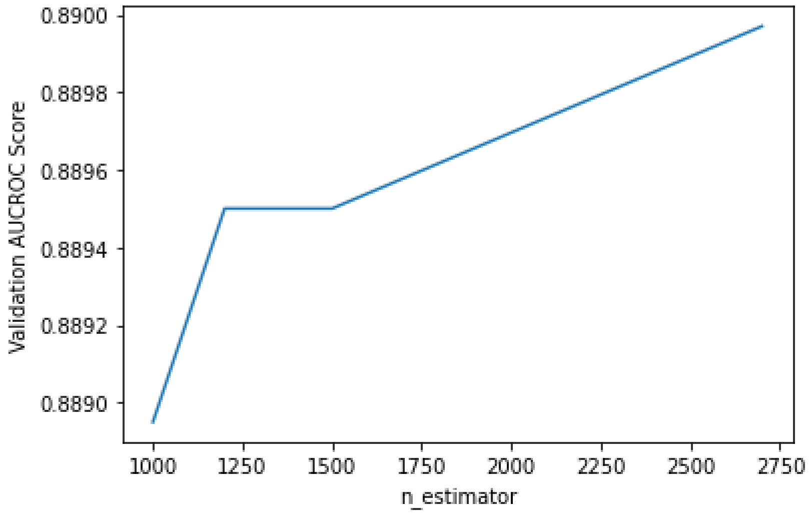
3. **VotingClassifier:** We used VotingClassifier on the basis of the following   models with different parameters

**Used models for voting classifier:**
2 Random Forest, 1 KNN, 2 Bagging Classifier, 1 ExtraTreesClassifier
Testing = **0.8749993613155925**

## Results:

By using different methodologies as mentioned above we found XGBoost is performing well So we plotted this model's performance on the different hyperparameters as shown in the following plot:





**We analyzed the above plots and found the highest test and train roc_auc scores at n_estimators=2700.**

**Finalized Models:** We tried different Models in this classification and tunned them on the different hyperparameters and found the following three models performed well on the testing dataset.

**Model 1:**
(objective="binary:logistic",learning_rate =0.01, n_estimators=1900, max_depth=12, min_child_weight=1, gamma=0, subsample=0.8, colsample_bytree=0.8, nthread=4, scale_pos_weight=1, seed=27)
**Training = 0.88720**
**Testing = 0.89017**

*** These scores are based on probability submission.**

**Model 2:**
(objective="binary:logistic",learning_rate =0.01, n_estimators=2800, max_depth=11, min_child_weight=1, gamma=0, subsample=0.8, colsample_bytree=0.8, nthread=4, scale_pos_weight=1, seed=30)
**Training = 0.89013**
**Testing = 0.89082**

*** These scores are based on probability submission.**

**Model 3:**
(objective="binary:logistic",learning_rate =0.01, n_estimators=1400, max_depth=14, min_child_weight=1, gamma=0, subsample=0.8, colsample_bytree=0.8, nthread=4, scale_pos_weight=1, seed=27)
**Training = 0.80170**
**Testing = 0.80033**

*** These scores are based on discrete class label submission.**

**We have submitted the above 3 models on the following drive link.**

**Link for models and CSV files:**
**https://drive.google.com/drive/folders/11K6sCenCvW6WCBig-iGM9hwoUU3oaf6E?usp=sharing**

**Learning:** In this assignment, we tried different models and tunned them on the different hyperparameters for improving the performance on the given classification dataset. The learning process is summarized below:

A. How to analyze the given dataset using Heatmap and if feature deduction is required or not?
B. How to handle imbalanced dataset by using different Sampling techniques like RandomOverSampling, RandomUnderSampling, and Smote?

C. How to perform Outlier detection by using Isolation Forest (Outlier detection is not included in submitted files because it wasn't increasing the performance of models)?
D. How to perform normalization/scaling the dataset. (Did not work well).
E. We got to know different Tree-based classifiers and their Hyperparameters tuning.
F. How to evaluate a machine learning model by using cross-validation and held out samples?
G. How to plot different types of count plot and linear plots using the seaborn library?

**--End--**