Predict Segment

- Team Members:

name	ID
Norhan Mahmoud Mohamed	2018170833
Amira Mohmed Gomaa	2018170721
Ganna Ayman Esmail	2018170729
Esraa Mohamed Ali	2018170713
Yomna Abdelsamed Abdelaal	2018170844

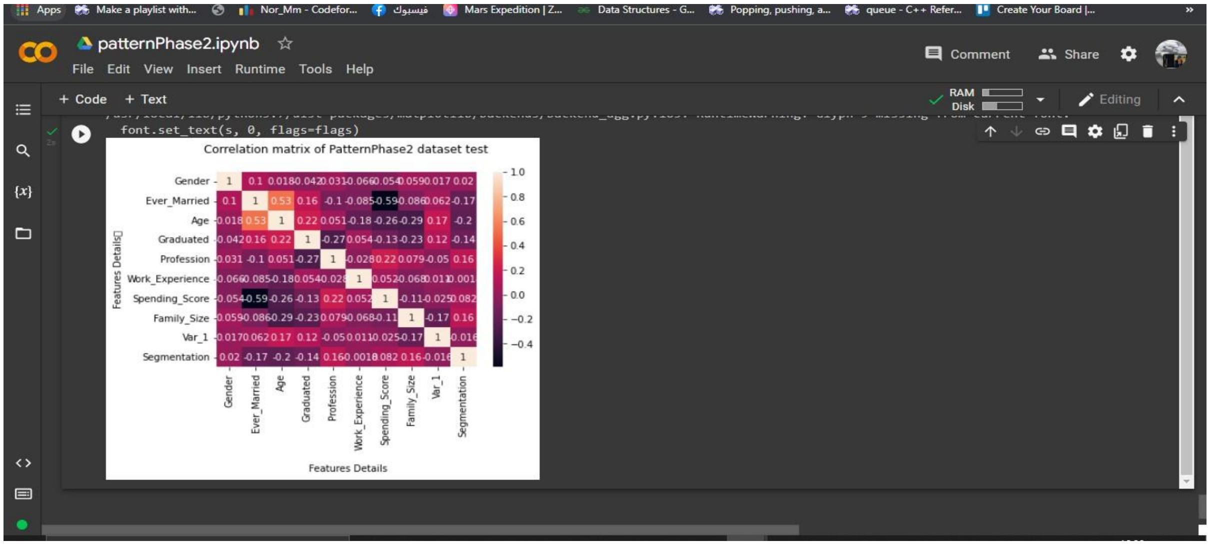
1. Target

Classify between 4 segment classes (A, B, C, D).

1

2- Correlation between Columns:

Train DataSet:



```
font.set_text(s, 0, flags=flags)
```

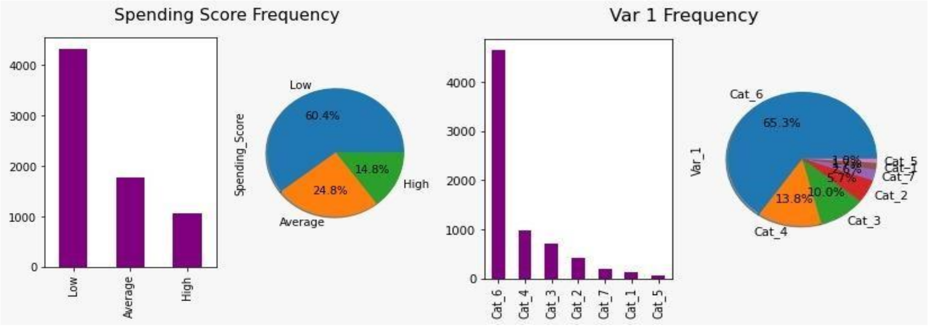
Correlation matrix of PatternPhase2 dataset test

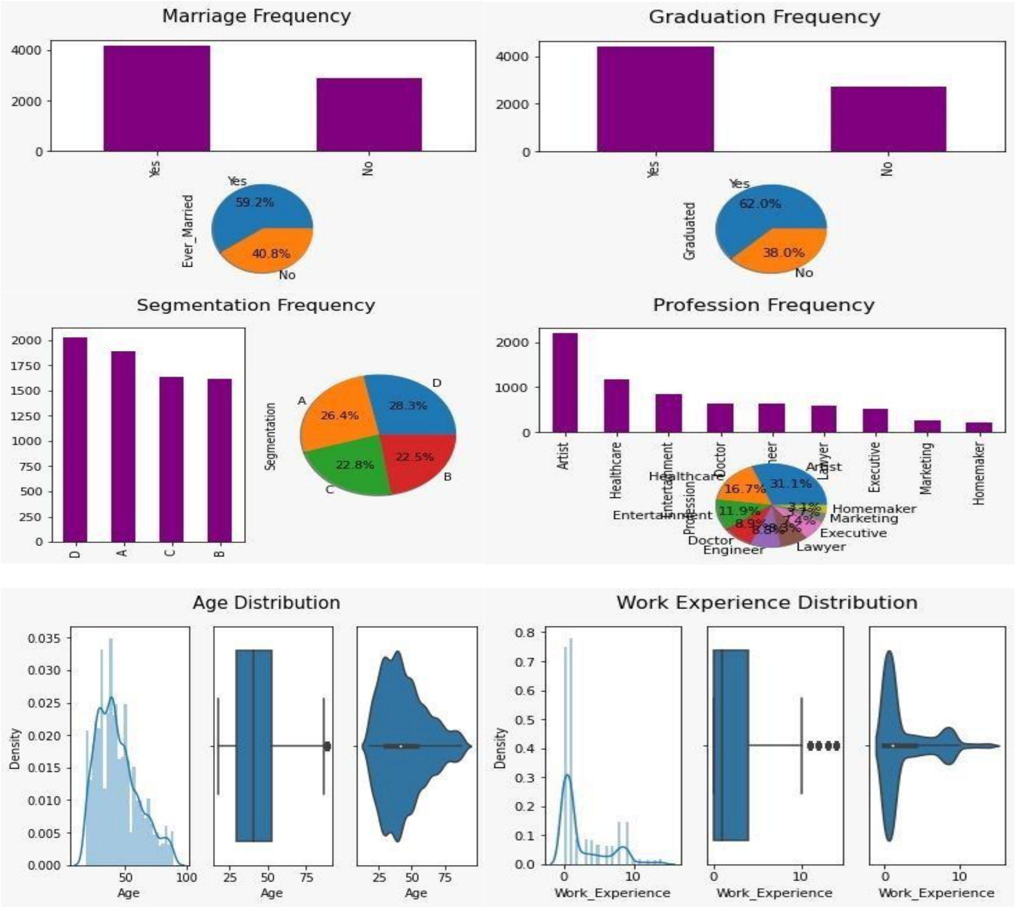
Features Details

Features Details	Gender	Ever_Married	Age	Graduated	Profession	Work_Experience	Spending_Score	Family_Size	Var_1	Segmentation
Gender	1	0.1	0.018	0.042	0.031	-0.066	0.054	0.059	0.017	0.02
Ever_Married	0.1	1	0.53	0.16	-0.1	-0.085	-0.59	0.086	0.062	-0.17
Age	0.018	0.53	1	0.22	0.051	-0.18	-0.26	-0.29	0.17	-0.2
Graduated	0.042	0.16	0.22	1	-0.27	0.054	-0.13	-0.23	0.12	-0.14
Profession	0.031	-0.1	0.051	-0.27	1	-0.028	0.22	0.079	-0.05	0.16
Work_Experience	0.066	0.085	-0.18	0.054	-0.028	1	0.052	0.068	0.011	0.001
Spending_Score	0.054	-0.59	-0.26	-0.13	0.22	0.052	1	-0.11	-0.025	0.082
Family_Size	0.059	0.086	-0.29	-0.23	0.079	0.068	-0.11	1	-0.17	0.16
Var_1	0.017	0.062	0.17	0.12	-0.05	0.011	0.025	-0.17	1	0.016
Segmentation	0.02	-0.17	-0.2	-0.14	0.16	0.0018	0.082	0.16	-0.016	1

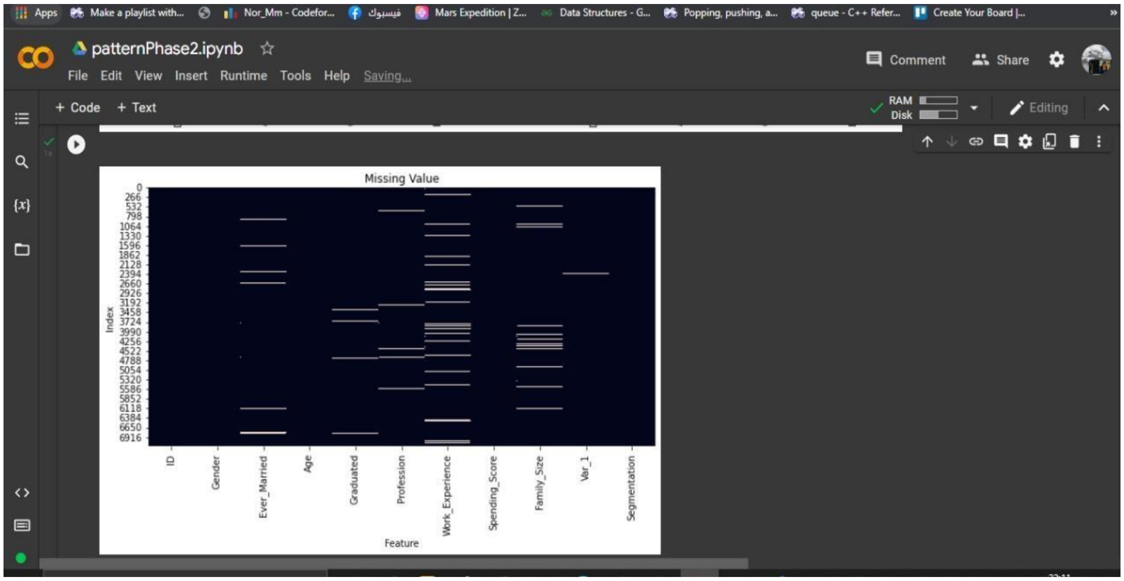
3-Data Analysis:

Feature	Description	Values	Null Value
Id	Unique ID		Without
Gender	Gender of the customer	Male or Female	Without
Ever_Married	Marital status of the customer	Yes or No	With
Age	Age of the customer	Range [18,89]	Without
Graduated	Is the customer a graduate?	Yes or No	With
Profession	Profession of the customer	Doctor or Artist or Executive or Healthcare or Entertainment or Lawyer or Homemaker or Engineering	With
Work_Experience	Work Experience in years	Range [0,14]	With
Spending_Score	Spending score of the customer	Low or high or average	Without
Family_Size	Number of family members for the customer (including the customer)	Range [1,9]	With
Var_1	Anonymised Category for the customer	Cat_ from Range [1,7]	With
Segmentation	Customer Segment of the customer → (target)	A or B or C or D	Without





Graphs show percentage to all features



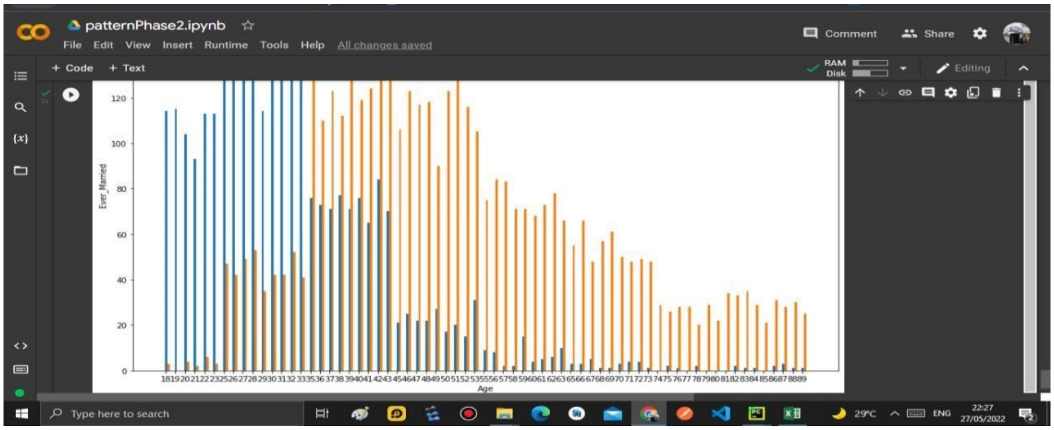
Graph shows Null Value

1- Age relate to Ever_Married feature ,When Age feature is less than 18 Ever_Married feature should be No, And when Age increase the Ever_Married Values (No) number decrease.

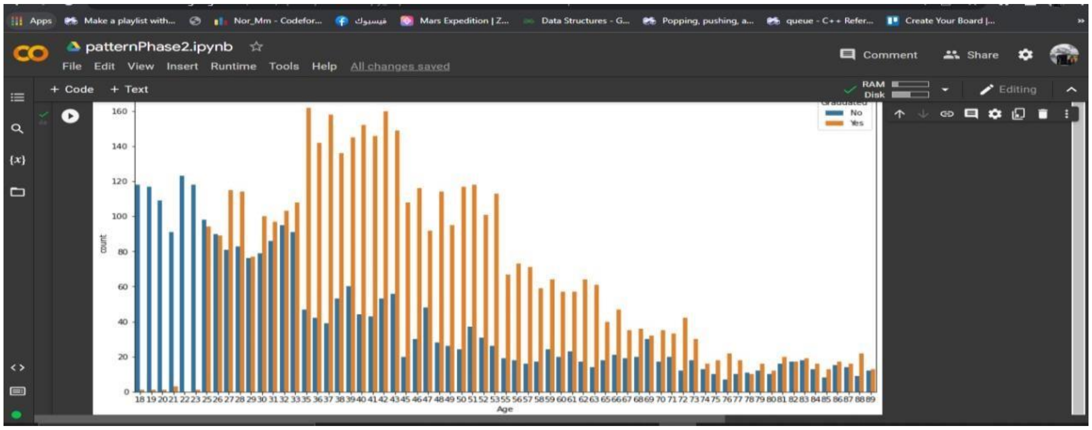
2- Age relate to Graduated feature , when age Is greater than 27 Graduated feature is yes.

3- Relation between graduation and profession feature, If graduation is no he should not to have any profession job.

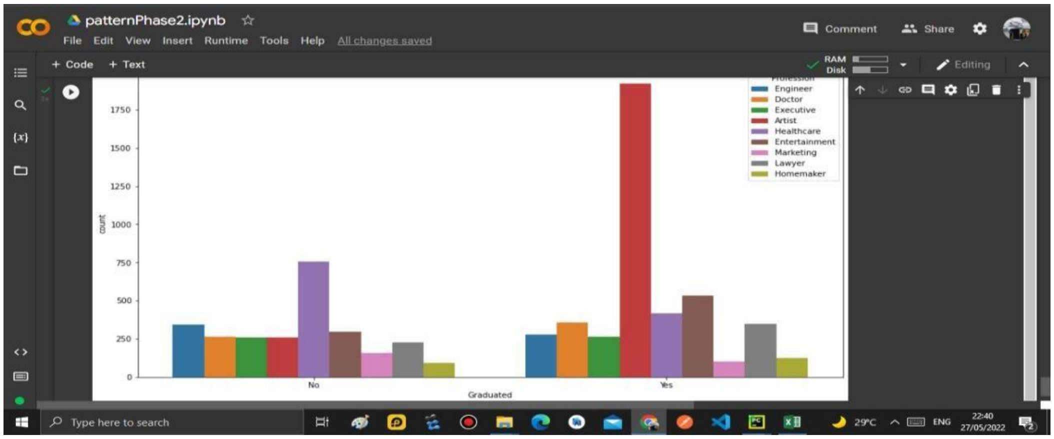
4- Segmentation related to Ever_Married and Profession feature.



Graph show relation between Age and Ever_Married



Graph showS relation between Age and graduated



Graph shows relation between profession and graduated feature

4- Pre-Processing On DataSet:

Train Data:

1- Drop Columns (ID & Segmentation).

2- Apply HotEncoding & LabelEncoding Columns (Gander , Ever_Married, Graduated)

3- Apply LabelEncoding Columns (Profession , Spending_Score , Var_1).

4- Try Fill Null Values with mean and most frequencies.

5- Try Remove Rows with Null value.

Test Data:

1- Drop Column (ID).
2- Fill Null value With mean and most frequencies.
3- Apply HotEncoding & LabelEncoding Columns (Gander , Ever_Married, Graduated).
4- 3- Apply LabelEncoding Columns (Profession , Spending_Score , Var_1).

Reduce Noise Values:

- ✓ When Age is greater than 27 , set graduated state to yes.
- ✓ When Age is less than 18 , set Married state to No.
- ✓ When graduated state is NO, set Profession state to Non Profession.
- ✓ Fill Null Value in Family Size Column by Linear Regression with Colums(Age & Ever_Marreid).

Data Split:

1- train 80% , test 20%.

2- K-fold (k=5).

5- Models:

1-Random Forest Model :

Build decision Tree Models and combine the result using majority voting.

-with accuracy = 0.47801814375436147.

Hyper-Paramters	Description
random_state =0	Control randomness.
n_estimators= 1800	Number of decision Tree Models.
max_depth= 4	Number of level from root to leaf.

2-AdaBoost Model:

higher points are assigned to the data points which are miss-classifiedor incorrectly predicted by the previous model. This means each successive model (random forest or decision tree) will get a weighted input.

-with accuracy = 0.466852756454989.

Hyper-Paramters	Description

random_state =0	Control randomness.
n_estimators= 100	Number of Models.
learning_rate= 0.1	how much to change.

3-Decision Tree Model:

learning simple decision rules inferred from the data features. A tree can be seen as a piecewise constant approximation. is a class capable of performing multi-class classification on a dataset.

-with accuracy = 0.4569832402234637.

Hyper-Paramters	Description
min_samples_leaf=40	Number of sample in leaf node
ccp_alpha=0.000001	how much to change.
criterion='entropy'	Measure quality of split.
Max_Depth =2	Number of Layer.
max_features=5	Number of feature.

5-KNN:

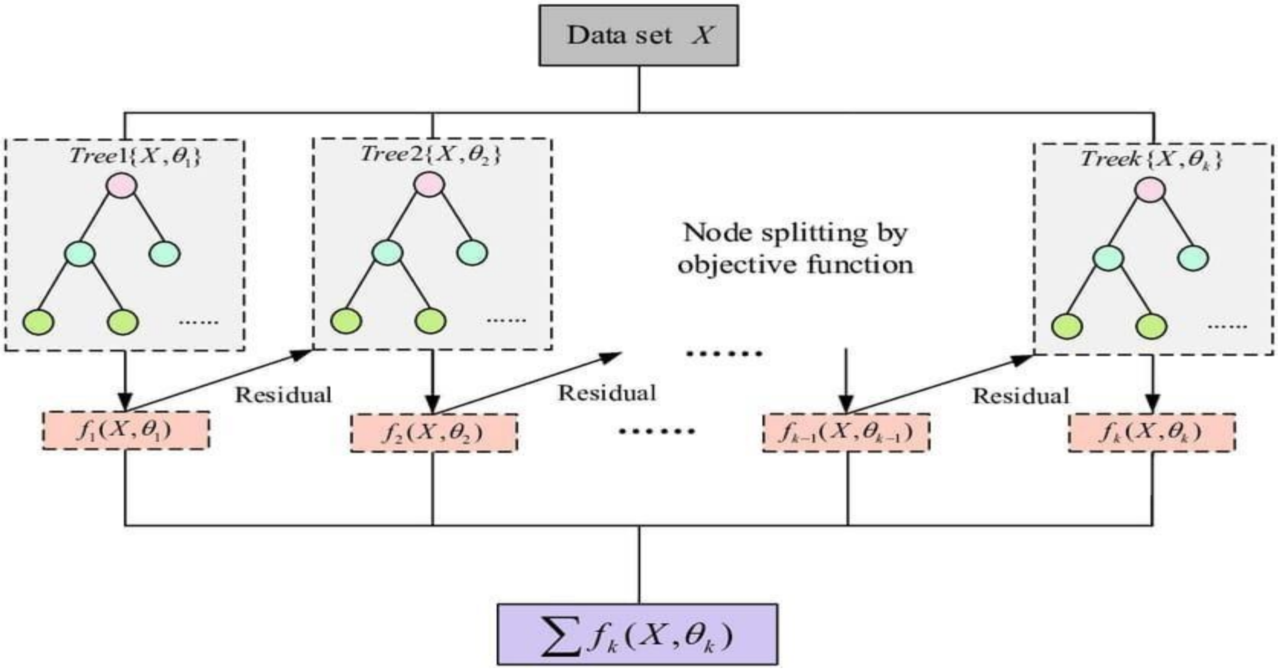
Apply majority voting to find the predicted class. It is also called a lazy learner algorithm because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset.

-with accuracy = 0.45638520586182835

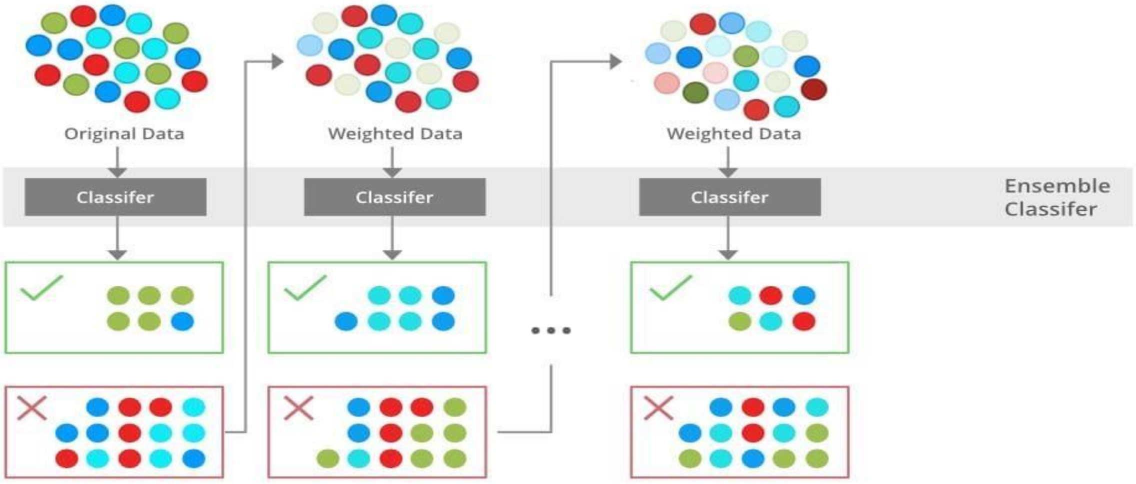
Hyper-Paramters	Description
n_neighbors = 35	Number of sample in leaf node
metric = 'minkowski'	decides the distance between the points.
p = 2	1 for Manhattan distance. 2 for Euclidean distance.

6- XGBoost:

XGBoost is an implementation of Gradient Boosted decision trees which regularize boosting and handle missing values automaticlly and enable early stoping.this algorithm use boosting algorithm. In boosting, the trees are built sequentially such that each subsequent tree aims to reduce the errors of the previous tree. Each tree learns from its predecessors and updates the residual errors. Hence, the tree that grows next in the sequence will learn from an updated version of the residuals.



Graph shows XGBoost Algorithm

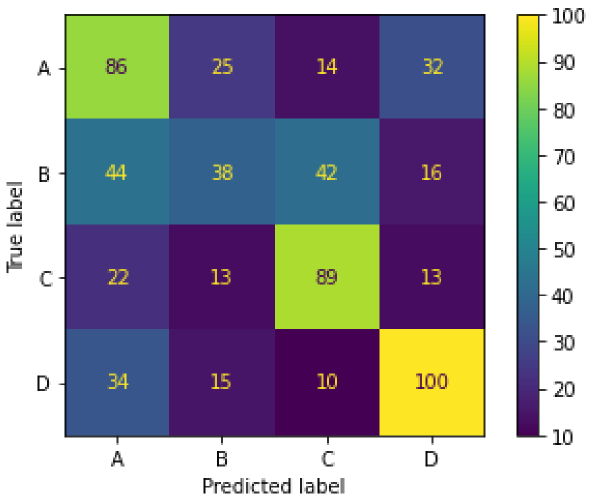


Graph shows boosting algorithm

2 Submissions:

in both submission we use XGBoost algorithm with different pre_processing.

1-first submission:



- **pre_processing :**

  -drop null data in training set.
  -drop columns (ID, Segmentation).
  -Labelencoding to all string feature.
  -Fill Null data in testing set by mean and most frequent.
  - k-fold (k=5).

  -When Age is greater than 27 , set graduated state to yes.

  -When Age is less than 18 , set Married state to No.

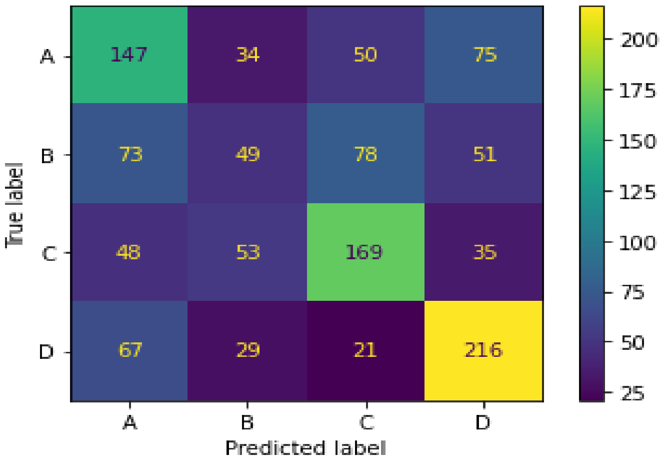
  -When graduated state is NO, set Profession state to Non Profession.

- **Hyper-Parameter:**

  objective="multi:softprob",random_state=0

-with accuracy =0.5278246205733558.

2-Secound Submission:



- **pre_processing :**

  -drop null data in training set.
  -drop columns (ID, Segmentation).
  -Labelencoding to all string feature.
  -Fill Null data in testing set by mean and most frequent.
  - k-fold (k=5).
  -

- **Hyper-Parameter:**

  objective="multi:softprob",random_state=0

-with accuracy = 0.486192468619246886.

Conclusion:

-When we drop feature it enhance accuracy with adaboost and decreases with xgboost algorithm.
-When we fill data with mean and median no change occur.

-When we deal with noise data it decrease accuracy.

-XGBoost algorithm give maximum accuracy between all algoritm.