Project Summary

Batch details	DSE Chennai October 2022
Team members	1. Ms. Anitha B 2. Mr. Manoj Prabhu S 3. Mr. Nawshath 4. Mr. Sai Charan 5. Mr. Suryakumar B 6. Ms. Vidhya Priya V
Domain of Project	Risk Analysis
Proposed project title	Innovative Approach on pandemic period: A Disciplinary Strategy
Group Number	7
Mentor Name	Mr. Mohit Sahu

Acknowledgement

We would like to thank our mentor Mr. Mohit Sahu for providing his valuable guidance and suggestions over the course of our Project Work. We also thank him for his continuous encouragement and interest towards our Project work.

We are extremely grateful to all our teaching and non-teaching staff members of GREAT LEARNING, who showed keen interest and inquired about our development and progress.

We greatly admire and acknowledge the constant support we received from our friends and team members for all the effort and hard work that they have put into completing this project.

TABLE OF CONTENTS

TABLE OF CONTENTS

1.INTRODUCTION

1.1 PROBLEM STATEMENT

The data from the medical industry has now been enormous and used in various platforms to analyze and predict multiple features. The prediction from those can either be targeted as medical or non-medical predictions. Number of days a patient stays in a hospital is a non-medical prediction that helps in deciding the risk of the patient and the availability of beds. But deciding the bed availability involves a lot of techniques and factors from the medical administration. The reason for this increased demand in bed availability also demands on various factors like Hospital type and severity of patients.

1.2 METHODOLOGY

The Methodology followed by the cross industry standard process for Data Mining (CRISP-DM). CRISP-DM provides the complete blueprint for conducting a Data Mining project. It provides a uniform framework for experience documentation.

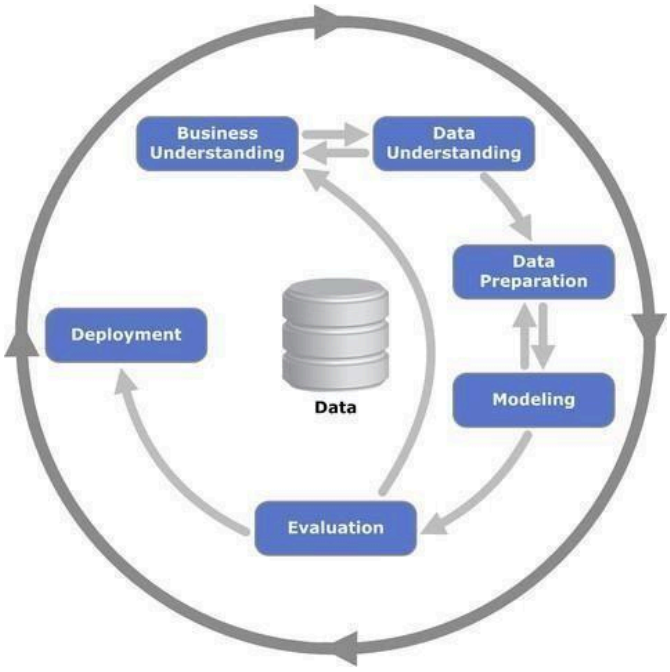


Fig 1.1 : Wikipedia

It breaks down the Life Cycle of a Data Mining project into the following phase

Phase 1 : Business Understanding

Understanding the business requirement is of paramount importance. The primary activity is to find out exactly what business is trying to accomplish. One needs to go deeper into fact finding to outline the issues in the business goals task.

After understanding the requirements, the next steps is to lay down every step that the data scientist intends to take until the project is accomplished and results are presented and reviewed.

Phase 2 : Data Understanding

The data understanding phase goes hand in hand with the business understanding phase. In this phase, the data will be collected, described, explored and verified. Data Understanding phase is where Exploratory Data Analysis is performed.

Phase 3 : Data Preparation

The third phase is Data Preparation which is also called as Data Wrangling or Data Munging Phase. It involves developing the final dataset for modeling. It covers all activities to construct the final dataset from the initial raw data. In this phase Data Cleaning, Imputation and Feature Engineering are performed

Phase 4 : Modeling

The fourth phase is Modeling, which involves selecting the actual modeling technique that needs to be used. In this phase the Modelling technique will be selected, test design will be generated, and a model will be built. Several models will be built based on tuning iterations until the best model is found.

Phase 5 : Evaluation

The fifth phase is Evaluation, where the model will be evaluated. The steps executed to construct the model to ascertain it properly achieves the business objectives.

Phase 6 : Deployment

The last and final phase is Deployment. However for this project, this step will not be executed.

2. DATASET AND DOMAIN

2.1 DATASET SOURCE:

The data used in this project was retrieved from Kaggle which is used to analyze the COVID 19 Stay days prediction under risk analytics.

COVID-19 Hospitals Treatment Plan | Kaggle

2.2 DATA DICTIONARY:

The Dataset contains the information related to the features of the vehicle loans and account related details. The Dataset comprises 41 features and 2,33,154 data points

Variable Name	Variable Description
Case ID	Indicates unique identity number for the case
Hospital	Name of the Hospital
Hospital type	Types of Hospital
Hospital city	The City in which the Hospital is present
Hospital region	Region of the Hospital
Available extra rooms	Number of Extra rooms available in the Hospital
Department	Department overlooking the case | ['radiotherapy' 'anesthesia' 'gynecology' 'TB & Chest disease' 'surgery']
Ward type	Types of Ward
Ward facility	Ward Facility Types
Bed Grade	Condition of Bed in the Ward
Patient ID	Indicates unique identity number for the Patient
City Code	City Code for the patient
Type of admission	Admission Type registered by the Hospital | ['Emergency' 'Trauma' 'Urgent']
Illness severity	Severity of the illness recorded at the time of admission | ['Extreme' 'Moderate' 'Minor']
Patient visitors	Visitors visiting the patient

Variable Name	Variable Description
Age	Age category | ['51-60' '71-80' '31-40' '41-50' '81-90' '61-70' '21-30' '11-20' '0-10' '91-100']
Admission Deposit	Deposit at the Admission Time
Stay Days	Stay Days by the patient (target) | ['0-10' '41-50' '31-40' '11-20' '51-60' '21-30' '71-80' 'More than 100 Days' '81-90' '61-70' '91-100']

2.3 VARIABLE CATEGORIZATION:

Numerical Features	3
Categorical Features	15
Total No of Features	18

2.4 Redundant Columns:

After viewing the dataset we are dropping and merging some features which are irrelevant for the EDA purpose. The features are given below:

- Case ID
- Patient ID

3.Pre Processing Data Analysis

3.1 Null Value Imputation:

- There are **4644** null values in our dataset. These null values are from a columns

- BED_GRADE AND CITY_CODE_PATIENT

- Total % of null values in the dataset is **0.03% and 1.42%** respectively**.**
- We have removed the null values from the dataset since the percentage of null values are minimum and there are no patterns to fill the null values.
- This removal of null values dosen't affect our model.

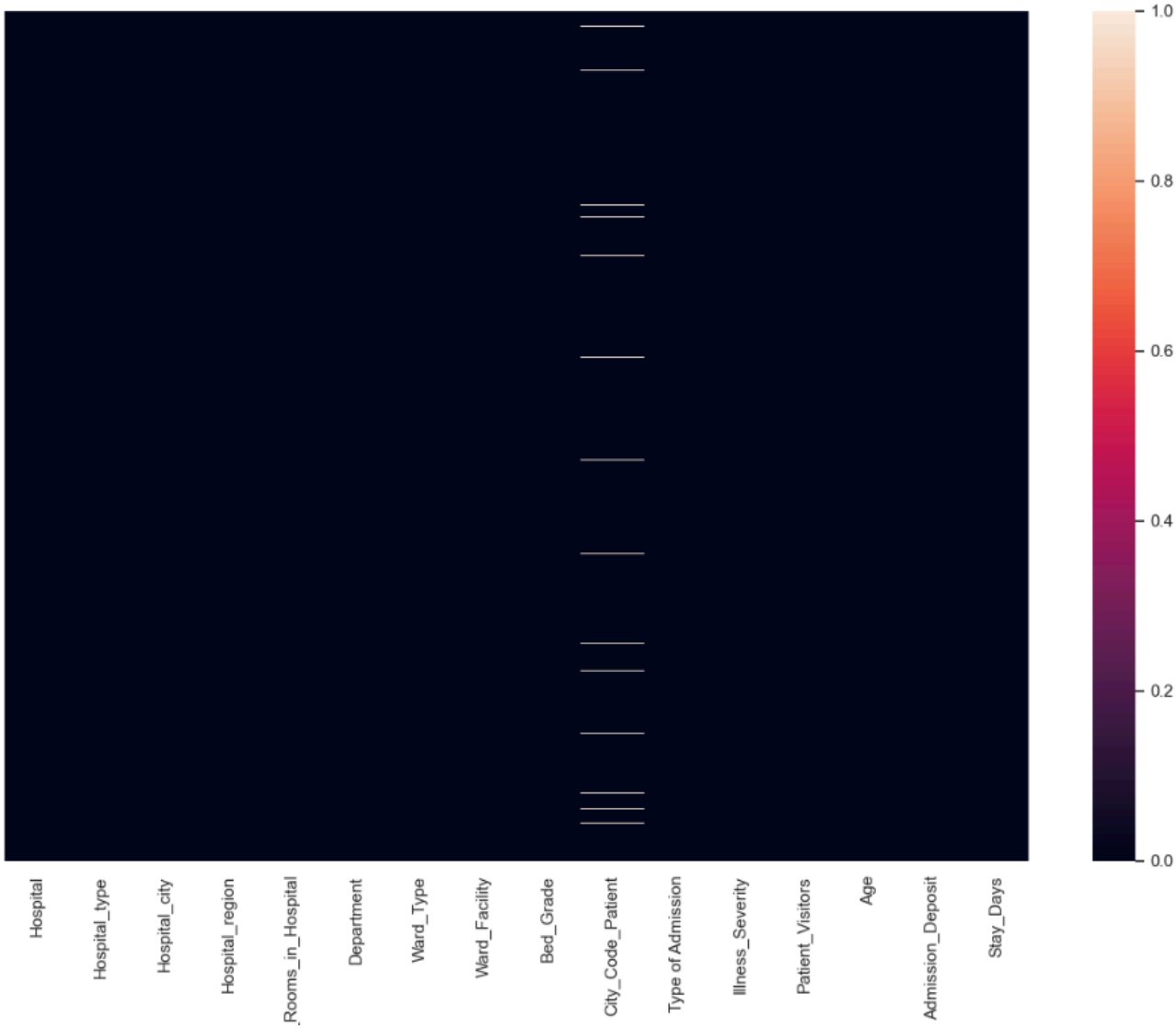


Fig 3.1 Representation of Null Values using Heatmap

4.EXPLORATORY DATA ANALYSIS

Exploratory Data Analysis refers to the critical process of performing initial investigations on data so as to discover patterns, to spot anomalies, to test hypotheses and to check assumptions with the help of summary statistics and graphical representations. Since numerous features are present in the dataset , certain features need to be dropped to proceed with feature engineering and exploratory data analysis privy to the missing value treatment which helps to gain more insights.

4.1 Relationship between variables

4.1.a UNIVARIATE ANALYSIS:

Univariate analysis is the simplest form of analyzing data. It doesn't deal with causes or relationships. It's major purpose is to describe; It takes data, summarizes that data and finds patterns in the data.

4.1.b TARGET VARIABLE(Dependent variable):

Before plotting the distribution, we need to encode it such that the target variable satisfies 1 for Yes and 0 for No.

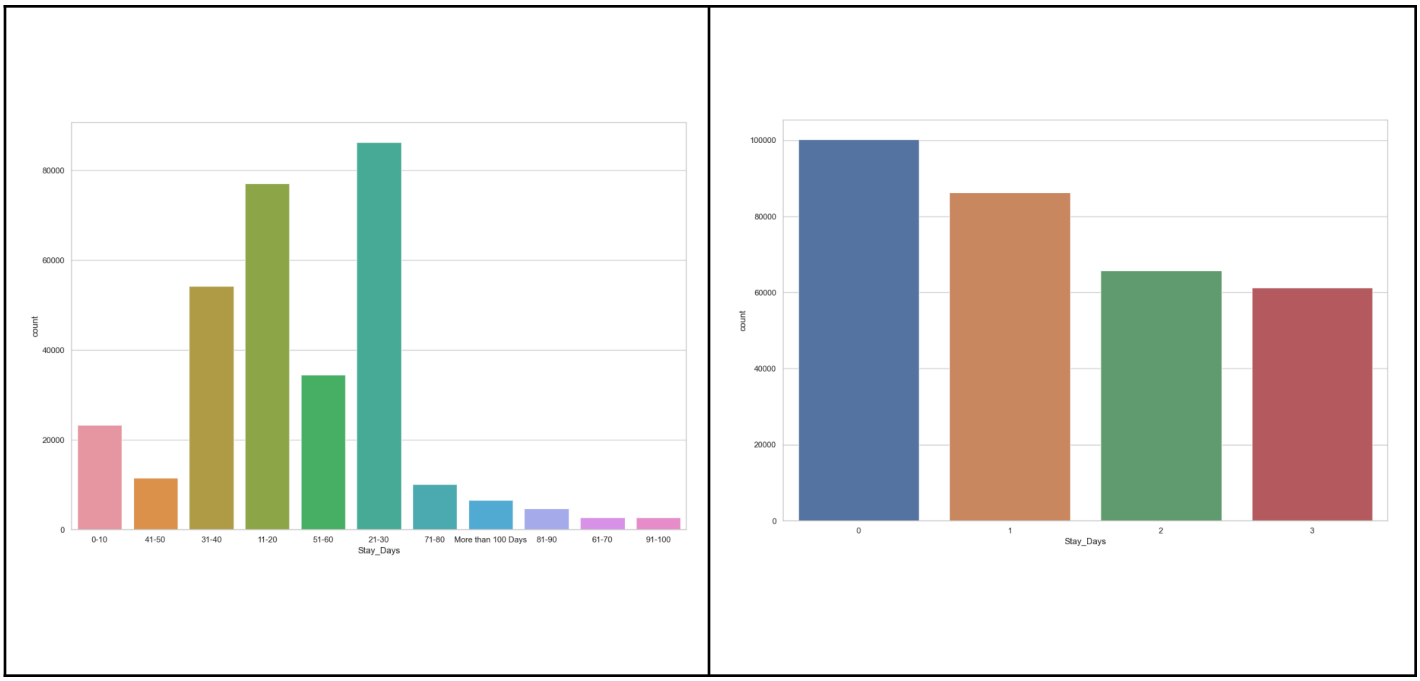


Fig 4.1:Target column - Whole data

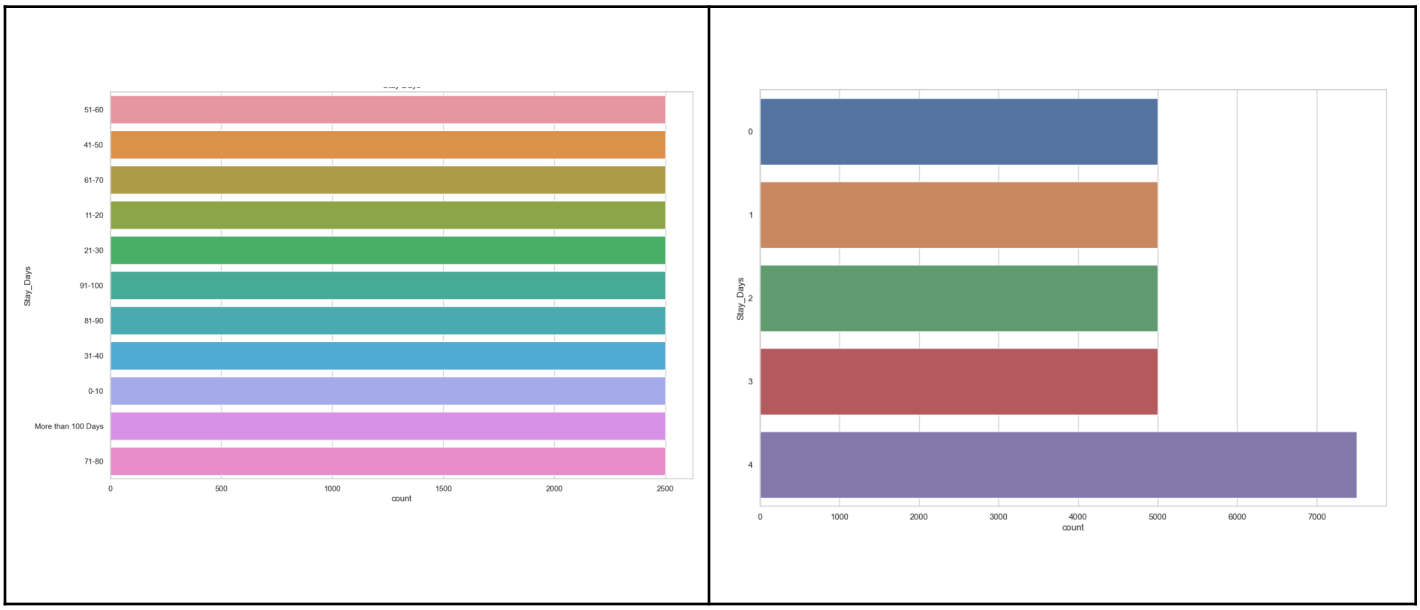
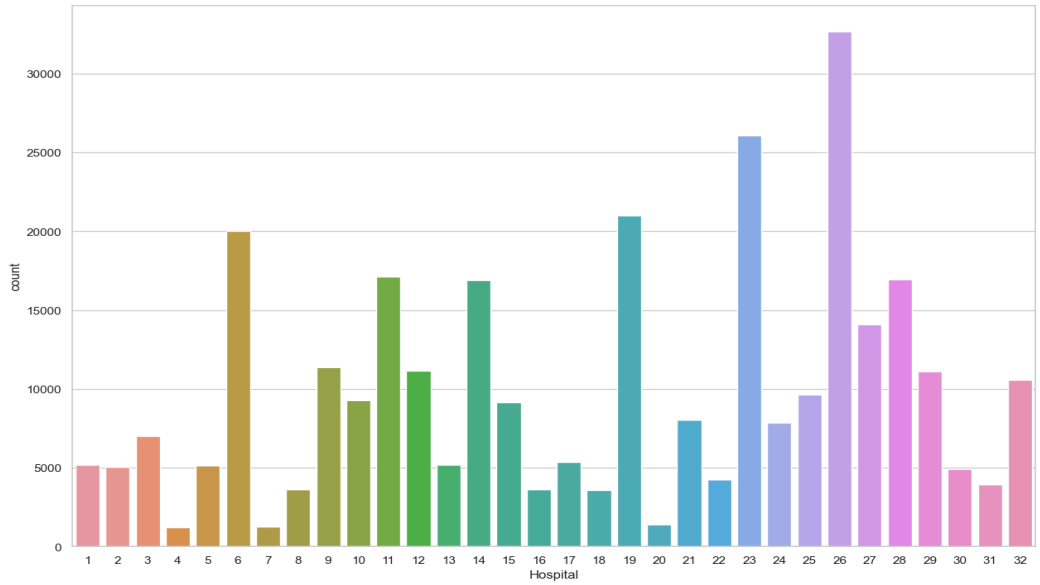


Fig 4.1.b:Target column -Binned data (Under-sampling)

4.1.C Uni-variate Analysis on Numerical Variable(Independent Variables):

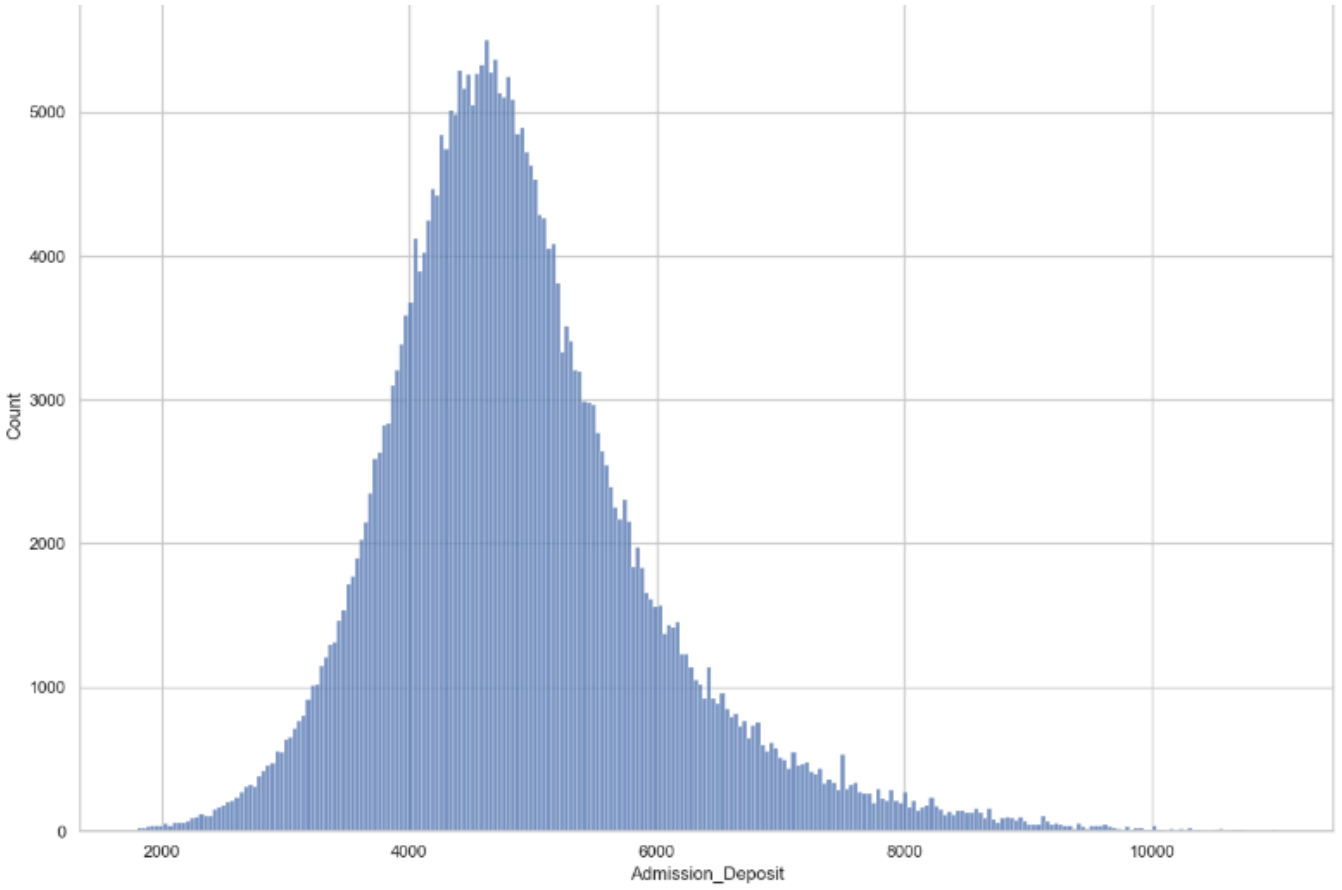
HOSPITAL:



Interpretation:

- Comparing to other hospitals, hospital 26 may have more number of patients details.
- Hospital 6,19,23 are may have patients details more than 20000 but lesser than hospital 26.
- Hospital 4,7,8,16,18,20,22,30,31 are may have patients details lesser than 5000.

AMOUNT DEPOSITED:

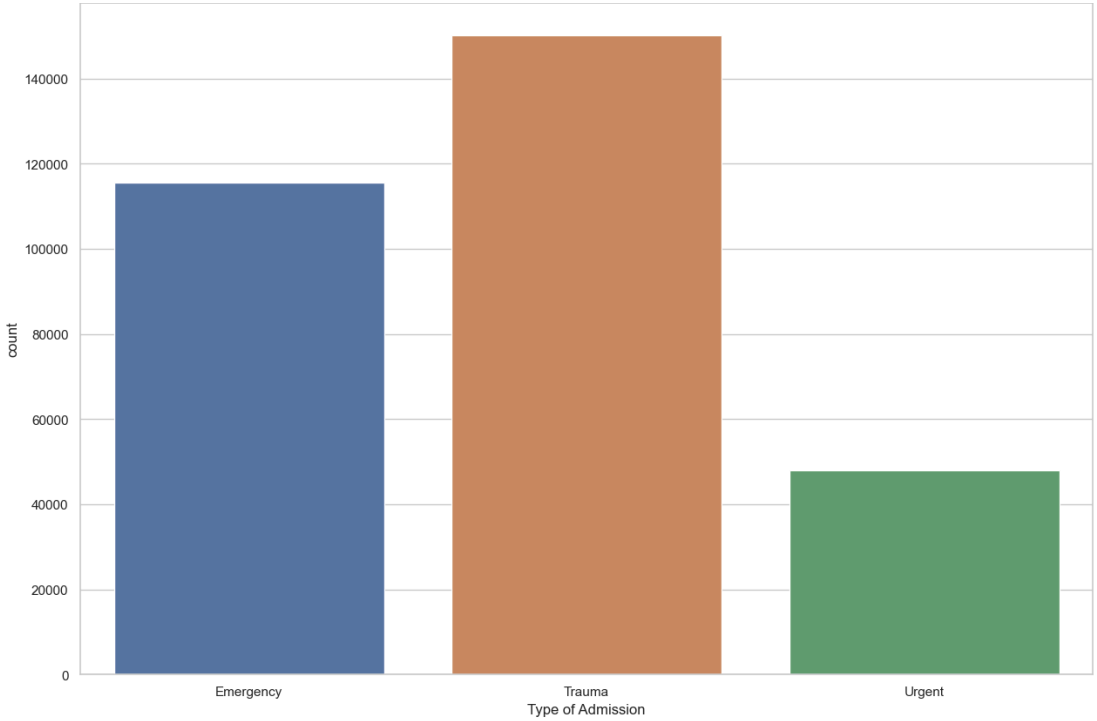


Interpretation:

- The amount deposited at the time of admission of a patient shows that the data is normally distributed.
- But still data shows slightly right skewness.

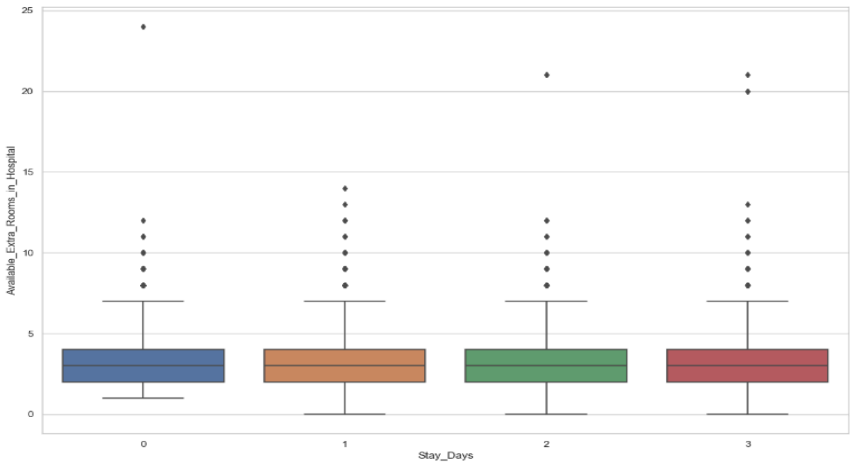
4.1.D Uni-variate Analysis on Categorical Variables:

TYPE OF ADMISSION:



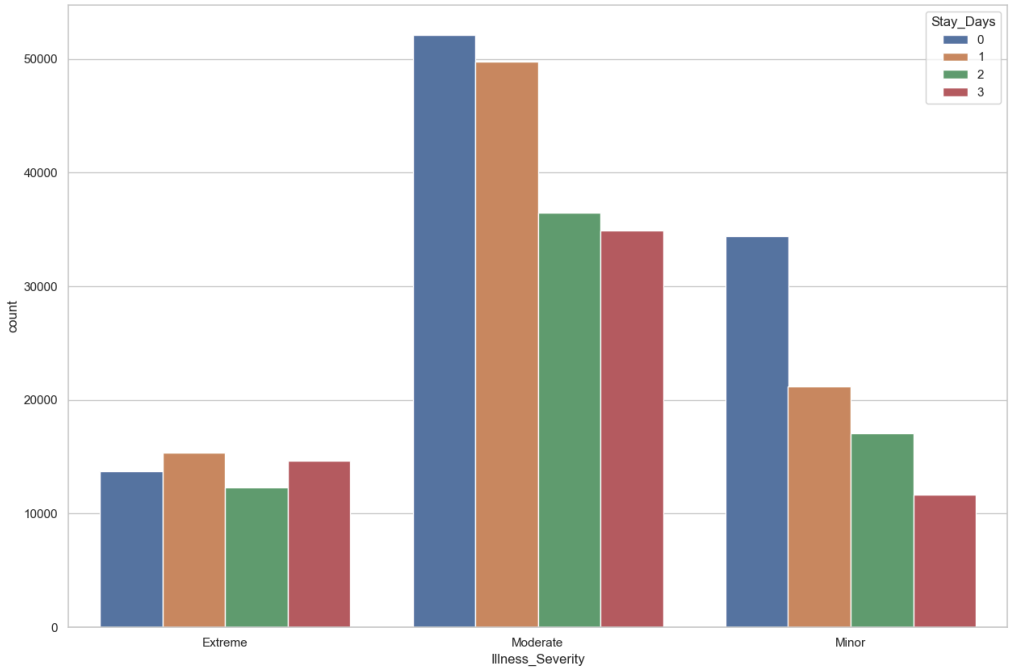
4.2.A Bivariate Analysis of Numerical Variable:

Availabe_extra_rooms:

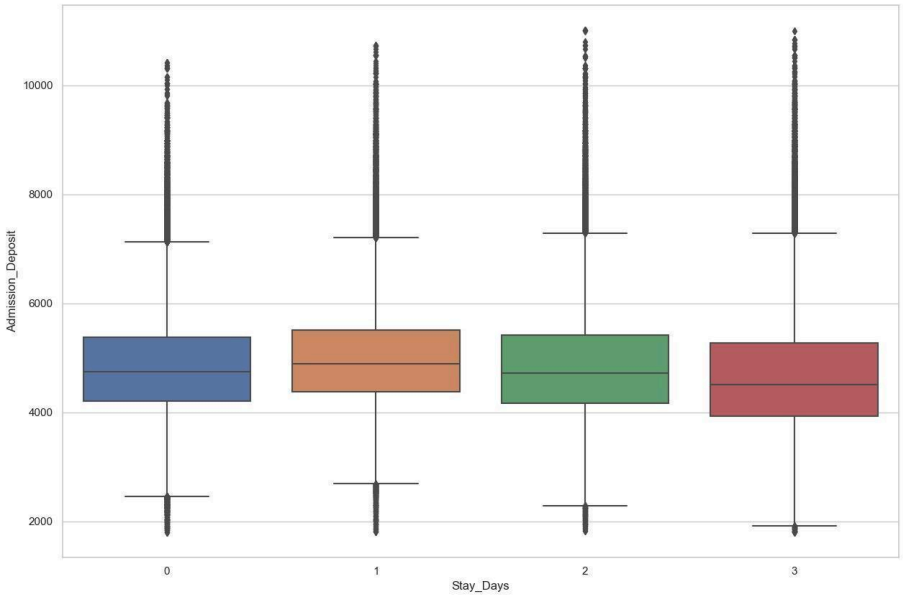


4.2.B. Bivariate Analysis on Categorical Variables:

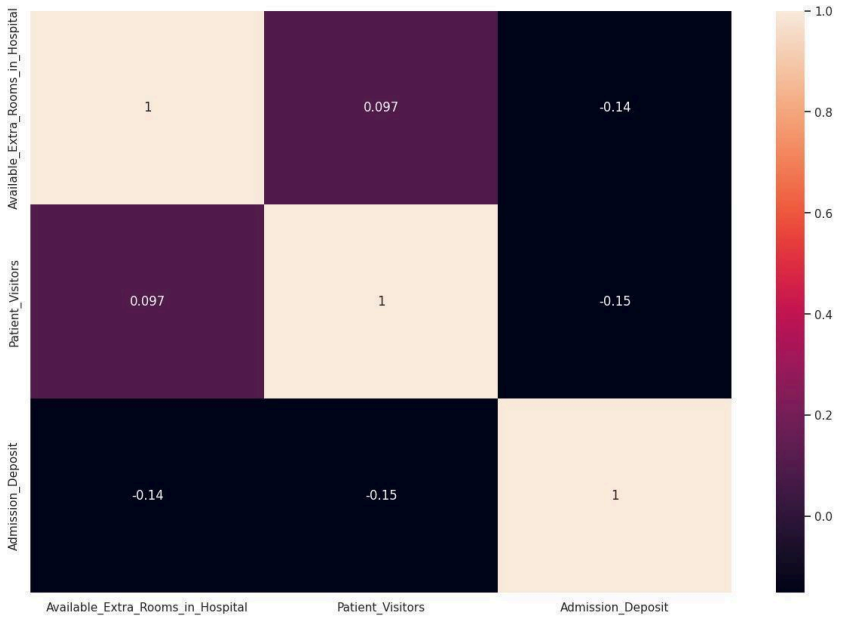
ILNESS SEVERITY:



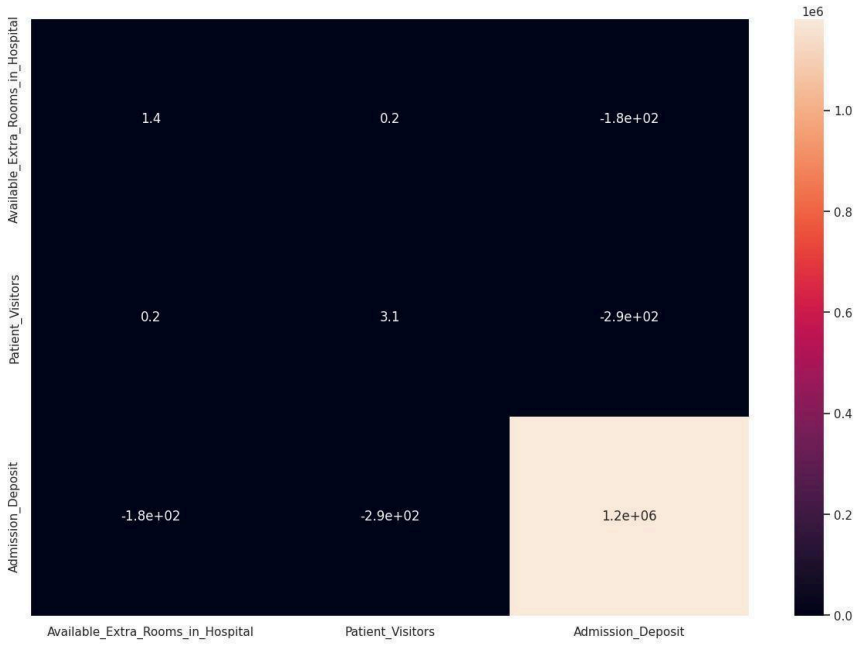
Amount_Deposited:



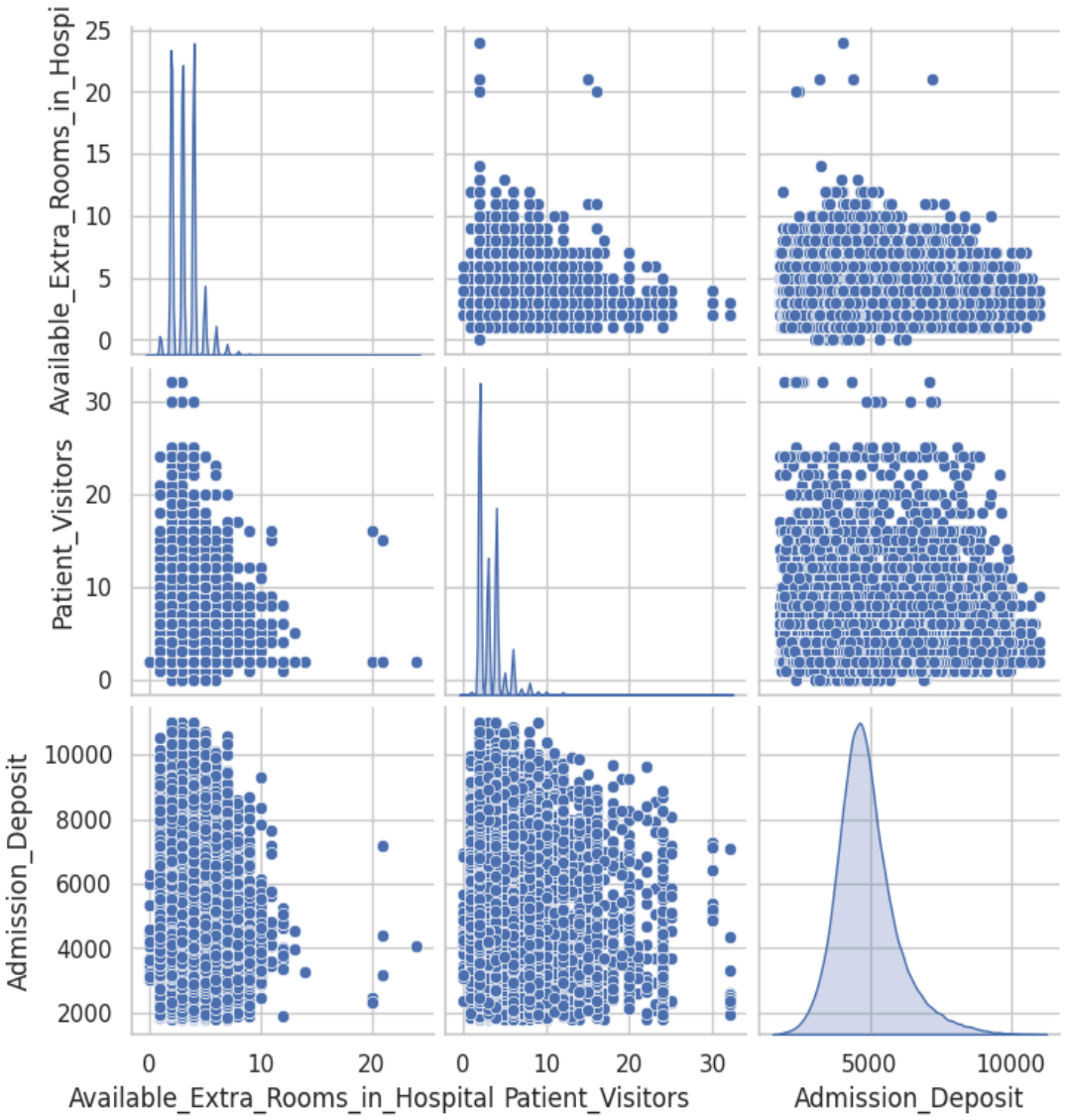
4.3.A Correlation between numerical variables:



4.3.B Covariance between variables:



4.4 Multivariate Analysis:



5.FEATURE ENGINEERING

5.1. ENCODING

We have encoded target column of the dataset in 3 different ways:

1. Overall target variable has been binned to 4 classes
2. Under-sampling was carried and encoded as it is
3. Under-sampling was carried and target variable binned into 5 classes

Other categorical columns: Illness_severity and Age are encoded ordinally, rest are encoded using One Hot encoding.

Feature Encoding

```
df['Illness_Severity'].replace({'Extreme':2,'Moderate':1,'Minor':0},inplace=True)

df.Age.replace({'0-10':0,'11-20':1,'21-30':2,'31-40':3,'41-50':4,'51-60':5,'61-70':6,'71-80':7,'81-90':8,'81-90':9,'91-100':10},inplace=True)

df.Stay_Days.replace({'0-10':0,'11-20':0,'21-30':1,'31-40':2,'41-50':2,'51-60':3,'61-70':3,'71-80':3,'81-90':3,'91-100':3,'More than 100 Days':3},inplace=True)

pd.options.display.max_columns=None

dummy_encode=pd.get_dummies(df,columns=['Hospital','Hospital_city','Hospital_region','Hospital_type','Department','Ward_Type','Ward_Facility','Bed_Grade','City_Code_Patient','Type of Admission'])
```

Fig 3.2: Encoded for overall dataset

Feature encoding

```
df['Illness_Severity'].replace({'Extreme':2,'Moderate':1,'Minor':0},inplace=True)
df.Age.replace({'0-10':0,'11-20':1,'21-30':2,'31-40':3,'41-50':4,'51-60':5,'61-70':6,'71-80':7,'81-90':8,'81-90':9,'91-100':10},inplace=True)
df.Stay_Days.replace({'0-10':0,'11-20':1,'21-30':2,'31-40':3,'41-50':4,'51-60':5,'61-70':6,'71-80':7,'81-90':8,'91-100':9,'More than 100 Days':10},inplace=True)

dummy_encode=pd.get_dummies(df,columns=['Hospital','Hospital_city','Hospital_region','Hospital_type','Department','Ward_Type','Ward_Facility','Bed_Grade','City_Code_Patient','Type of Admission'])
```

Fig 3.3: Encoded for undersampling dataset

Feature encoding

```
df['Illness_Severity'].replace({'Extreme':2,'Moderate':1,'Minor':0},inplace=True)
df.Age.replace({'0-10':0,'11-20':1,'21-30':2,'31-40':3,'41-50':4,'51-60':5,'61-70':6,'71-80':7,'81-90':8,'81-90':9,'91-100':10},inplace=True)
dummy_encode=pd.get_dummies(df,columns=['Hospital','Hospital_city','Hospital_region','Hospital_type','Department','Ward_Type','Ward_Facility','Bed_Grade','City_Code_Patient','Type of Admission'])
dummy_encode.head()
```

Fig. 3.4: Encoded for undersampled target binned

5.2 SCALING:

Feature Scaling needs to be performed when dealing with Gradient Descent Based algorithms (Logistic Regression model for this data) and Distance-based algorithms as these are very sensitive to the range of the data points. This step is not mandatory when dealing with Tree-based algorithms (like Decision Tree and Random Forest) and for Boosting algorithms (like XGBoost).

Standardization was implemented to scale the data, where the data is transformed to have a mean of 0 and a standard deviation to 1.

Scaling the numerical features.

```
scaler=StandardScaler()
for i in num_data.columns:
  scaler.fit(dummy_encode[[i]])
  dummy_encode[i]=scaler.transform(dummy_encode[[i]])
dummy_encode
```

5.3. TRANSFORMATION:

Transformation is a process of converting data from one form to another to make it more suitable for analysis or modeling.Some common transformation techniques include scaling, normalization,feature extraction, and dimensionality reduction.

```
pt=PowerTransformer()
for i in num_data.columns:
  dummy_encode[i]=pt.fit_transform(dummy_encode[[i]])
dummy_encode
```

6.STATISTICAL ANALYSIS

6.1 Anova Test:

ANOVA (Analysis of Variance) is a statistical technique used to test whether the means of two or more groups are significantly different from each other. ANOVA is used to determine whether there is a statistically significant difference between the group means, and if so, which group or groups are responsible for the difference.

```
sig_lvl=0.05
for i in num_data.columns:
    print(i,'\n.........................................')
    result=df.groupby(by='Stay_Days')[i].apply(list)
    print('Hypothesis Framing')
    print('------------------')
    print('H0: The averages of all classes are the same.')
    print('H1: Atleast one class has a different average.\n\n')

    stat,pval=stats.f_oneway(*result)
    print('Pvalue= ',pval)
    if pval<sig_lvl:
        print('Result')
        print('------')
        print('pval<sig_lvl')
        print('Rejecting the null hypothesis')
        print('Atleast one class has a different average.')
        print('_____________________________________________________________________________________\n\n\n')
    else:
        print('Result')
        print('------')
        print('pval>sig_lvl')
        print('Failed to reject null hypothesis')
        print('The averages of all classes are the same.')
        print('_____________________________________________________________________________________\n\n\n')
```

Fig 6.1 Anova test

6.2 Chi Square Test:

Chi-square is a statistical test used to determine whether there is a significant association between two categorical variables. The Chi-square test is used to compare the observed frequencies of different categories with the expected frequencies, assuming that there is no association between the two variables.

```
for i in cat_data.columns:
    print('\n\n\n',i,'\n.................................\n\n\n')
    result = pd.crosstab(index=df['Stay_Days'],columns=df[i])
    chi2,pval,dof,exp=stats.chi2_contingency(result)
    print('Hypothesis Framing')
    print('------------------')
    print('H0: The variables are independent.')
    print('H1: The variables are not independent.\n\n')

    print('Pvalue= ',pval)
    if pval<sig_lvl:
        print('Result')
        print('------')
        print('pval<sig_lvl')
        print('Rejecting the null hypothesis')
        print('The variables are dependent.')
        print('_____________________________________________________________________________________________\n\n\n')
    else:
        print('Result')
        print('------')
        print('pval>sig_lvl')
        print('Failed to reject null hypothesis')
        print('The variables are independent.')
        print('_____________________________________________________________________________________________\n\n\n')
```

Fig 6.2 Anova test

The statistical analysis has to carried out for the whole data and under-sampled data. The results and the p-val are in the notebook.

7.APPLYING MACHINE LEARNING MODELS

7.1. Train Test Split:

The data is split into dependent feature Y (target) and independent features X. The data is then split into training and testing sets in order to avoid data leakage. The default 80:20 split is done.

7.2. LOGISTIC REGRESSION MODEL

Logistic regression is a statistical method used to analyze the relationship between a binary dependent variable (i.e., one that takes on one of two possible values) and one or more independent variables. It is a type of regression analysis that uses a logistic function to model the probability of the dependent variable taking on one of the two possible values as a function of the independent variables. The logistic function is an S-shaped curve that maps any real-valued input to a value between 0 and 1, which can be interpreted as the probability of the binary outcome. Here logistic regression has been carried out for multi classification model.

Model Evaluation:

1. **Accuracy score:**

It is the percentage of correct predictions made by the model out of all the predictions made. It is defined as the ratio of the number of correct predictions to the total number of predictions.

Accuracy score: 0.4950887797506611

2. **Confusion matrix:**

It is a table used to evaluate the performance of a classification model. It shows the number of true positives, false positives, true negatives, and false negatives for each class. The rows represent the actual values, while the columns represent the predicted values.

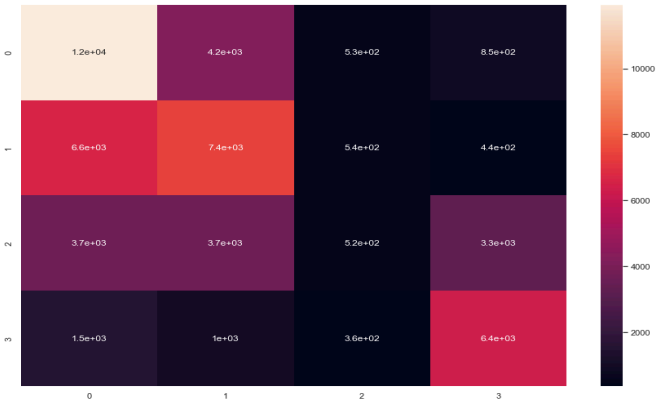


Fig. confusion matrix

3. Classification report:

It is a summary of the performance of a classification model for each class. It includes precision, recall, F1 score, and support (the number of samples in each class).

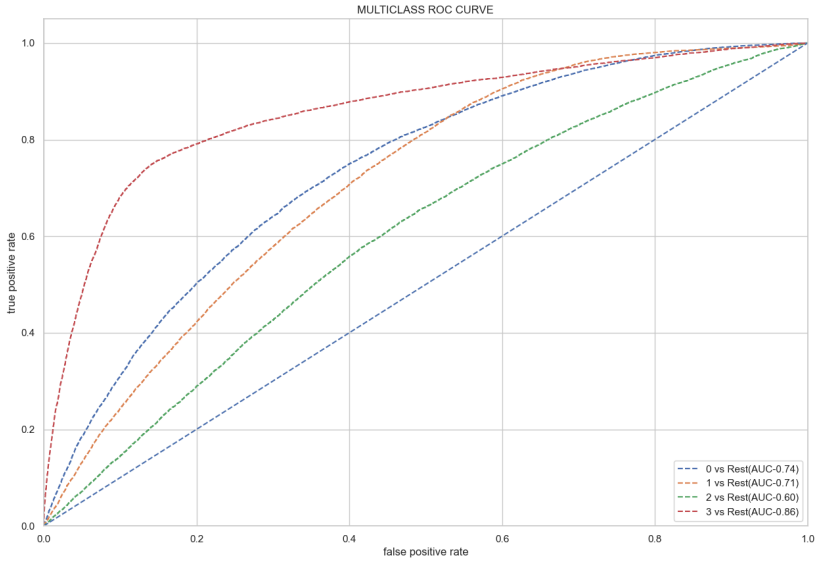
```
classification report:
              precision    recall  f1-score   support

           0       0.50      0.68      0.58     17529
           1       0.45      0.49      0.47     15001
           2       0.27      0.05      0.08     11219
           3       0.58      0.69      0.63      9191

    accuracy                           0.50     52940
   macro avg       0.45      0.48      0.44     52940
weighted avg       0.45      0.50      0.45     52940
```

4. ROC CURVE

ROC (Receiver Operating Characteristic) curve is a graphical representation of the performance of a binary classification model at different classification thresholds. It shows how well the model can distinguish between the positive and negative classes by varying the classification threshold.



7.3. DECISION TREE CLASSIFIER

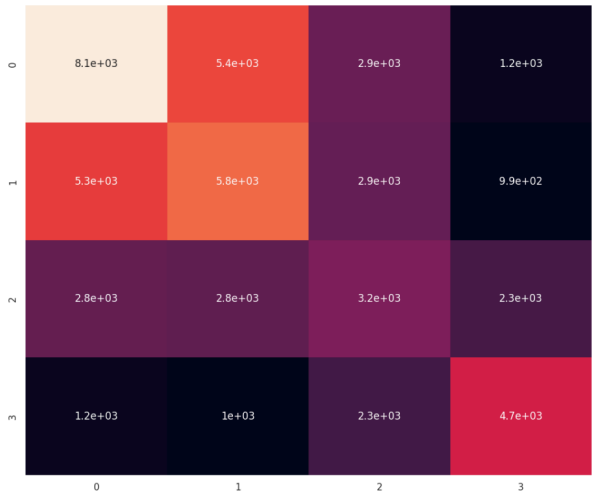
Decision tree is a machine learning algorithm that uses a tree-like model of decisions and their possible consequences to make predictions. It recursively splits the data into subsets based on the value of an attribute until it reaches a point where the decision can be made. Each internal node of the tree corresponds to an attribute, and each leaf node corresponds to a class label. Decision trees can be used for both classification and regression problems.

Train Dataset Evaluation:

1.Accuracy Score:

```
Accuracy score:  0.4136947487721949
```

2. Confusion matrix



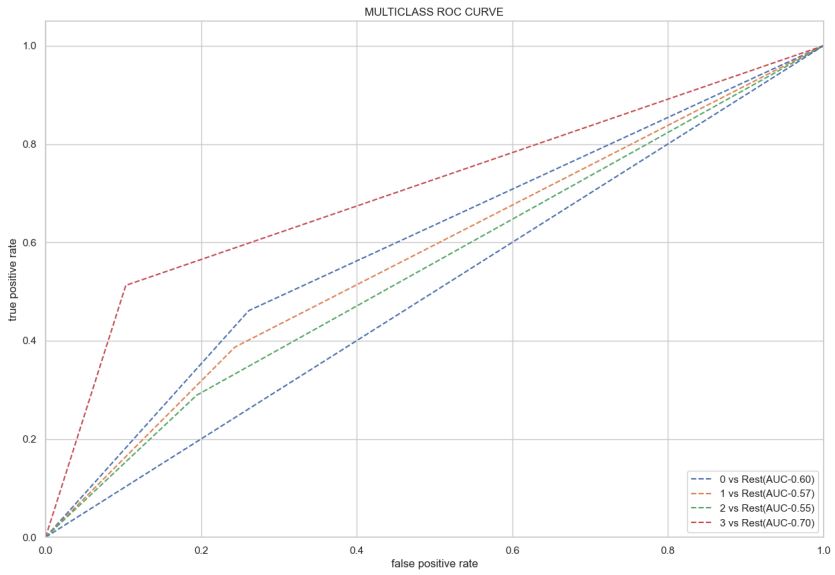
3. Classification Report

```
classification report:
              precision    recall  f1-score   support

           0       0.47      0.46      0.46     17529
           1       0.39      0.39      0.39     15001
           2       0.29      0.29      0.29     11219
           3       0.51      0.52      0.51      9191

    accuracy                           0.41     52940
   macro avg       0.41      0.41      0.41     52940
weighted avg       0.41      0.41      0.41     52940
```

4. ROC CURVE



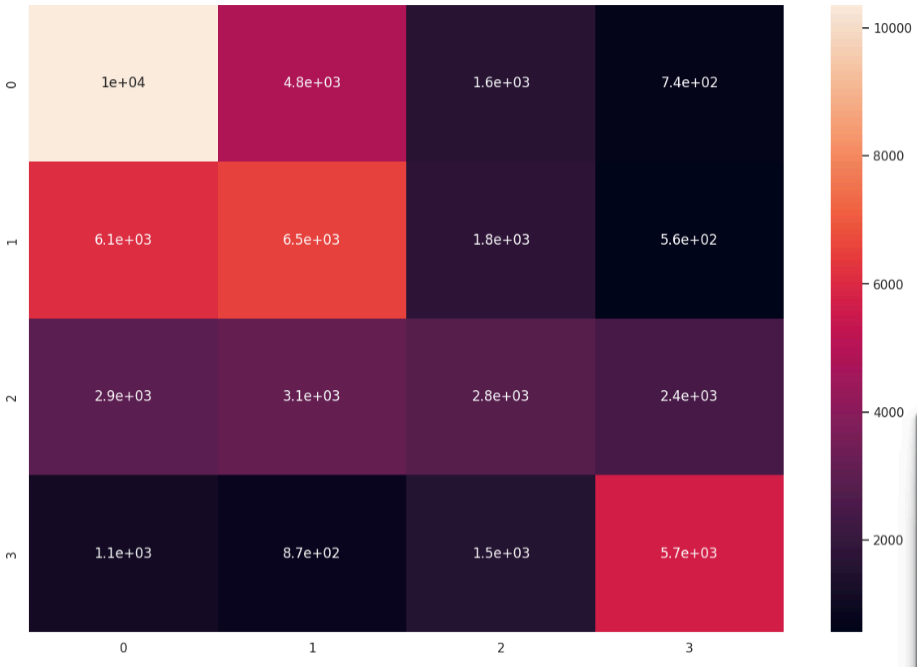
7.4. RANDOM FOREST CLASSIFIER MODEL

Random forest is an ensemble learning method that combines multiple decision trees to improve the accuracy and stability of predictions. It creates a forest of decision trees where each tree is trained on a random subset of the data and a random subset of the features. The final prediction is made by aggregating the predictions of all the trees in the forest.

1. **Accuracy Score:**

```
Accuracy score:  0.4798073290517567
```

2. Confusion matrix

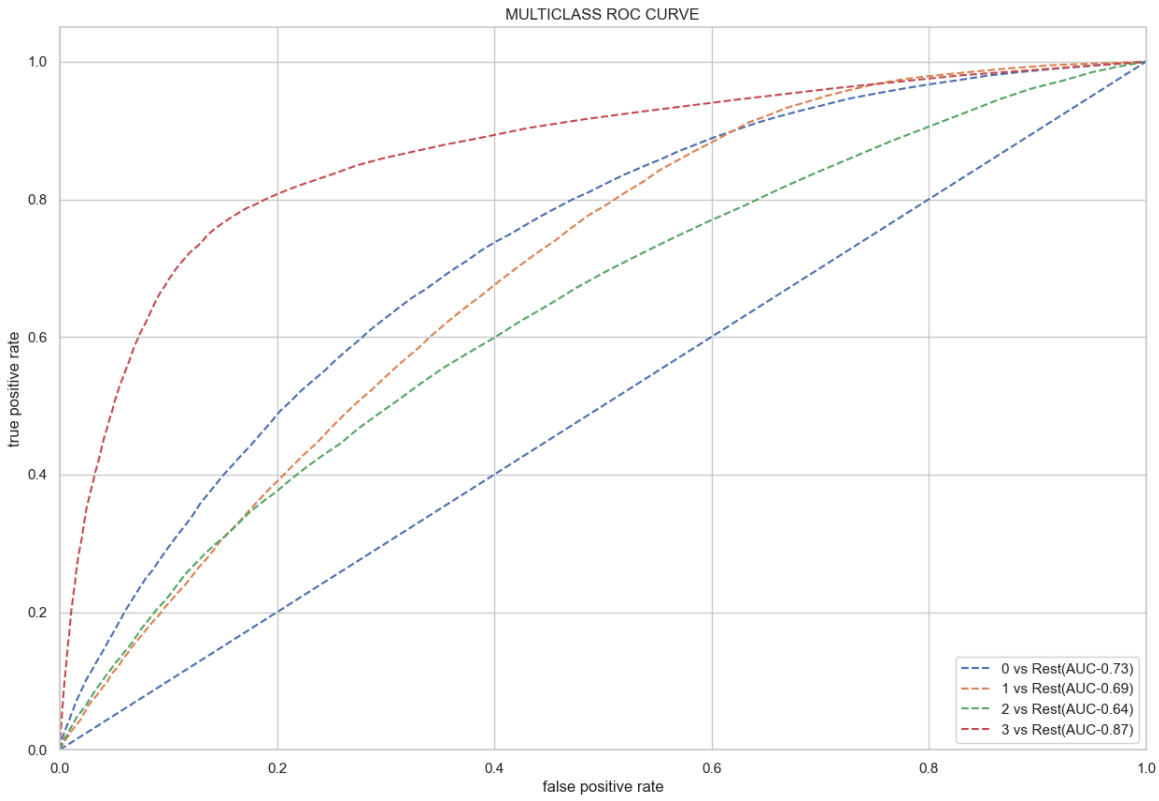


3. Classification Report:

classification report:

	precision	recall	f1-score	support
0	0.51	0.59	0.55	17529
1	0.43	0.43	0.43	15001
2	0.36	0.25	0.29	11219
3	0.61	0.62	0.62	9191
accuracy			0.48	52940
macro avg	0.47	0.47	0.47	52940
weighted avg	0.47	0.48	0.47	52940

4. ROC CURVE



7.5. ADA BOOST CLASSIFIER

BUILDING MODEL BASED ON ADA BOOST CLASSIFIER AND EVALUATION

Train Dataset Evaluation:

1. Accuracy Score:

The accuracy score is calculated with a calculated optimal threshold

```
Accuracy score:  0.48451076690593126
```

2. Classification Report:

classification report:	precision	recall	f1-score	support
0	0.51	0.65	0.57	17529
1	0.44	0.48	0.46	15001
2	0.30	0.14	0.19	11219
3	0.60	0.60	0.60	9191
accuracy			0.48	52940
macro avg	0.46	0.47	0.45	52940
weighted avg	0.46	0.48	0.46	52940

7.6 BAGGING CLASSIFIER

A Bagging Classifier is a type of ensemble learning algorithm that uses bootstrap aggregation (bagging) to combine the predictions of multiple base classifiers. The idea behind bagging is to create several random subsets of the training data and train a separate classifier on each subset

1. **Accuracy Score:**

The accuracy score is calculated with a calculated optimal threshold

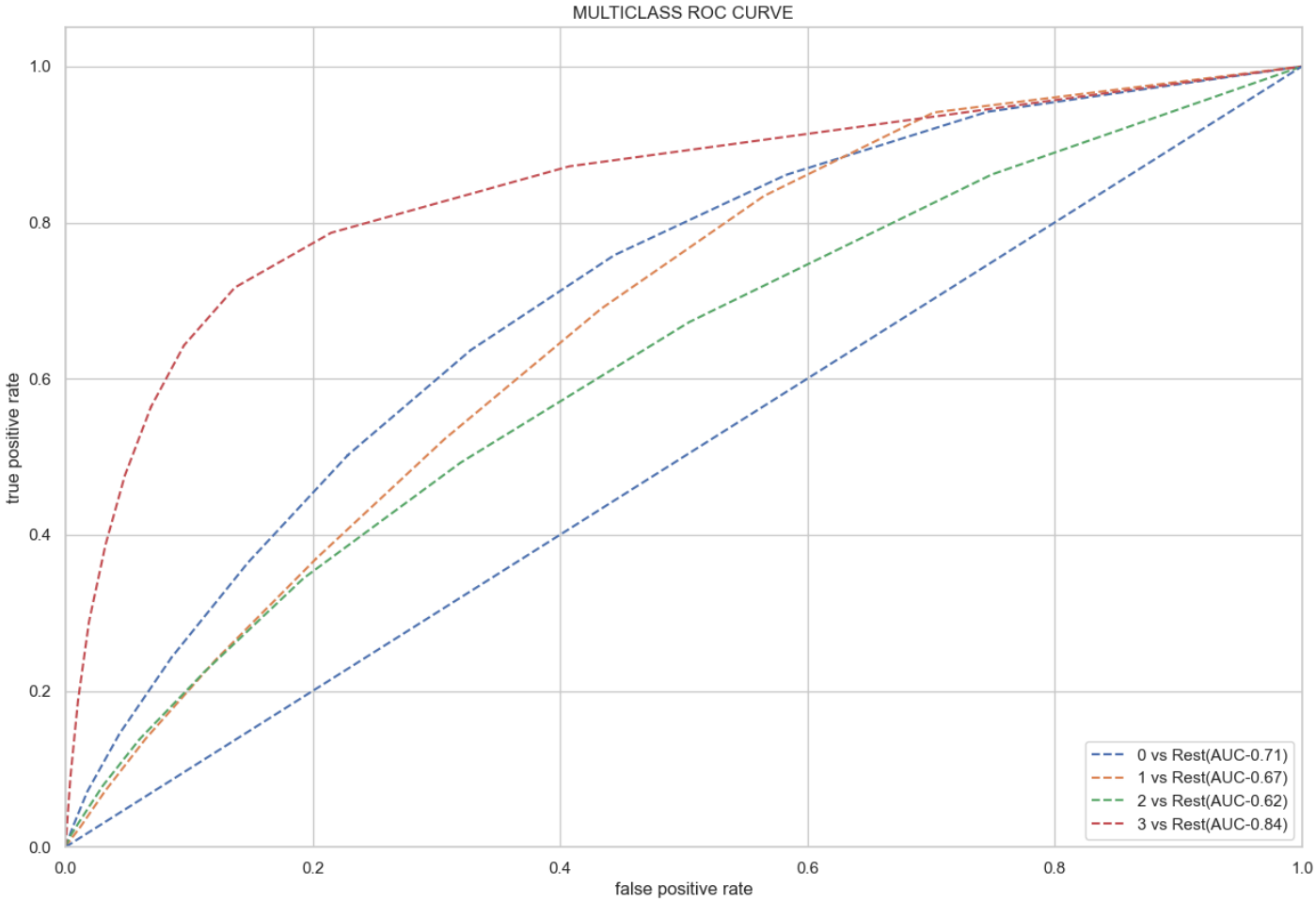
```
Accuracy score:  0.4671892708726861
```

2. Confusion matrix

3. Classification Report:

	precision	recall	f1-score	support
0	0.49	0.60	0.54	17529
1	0.42	0.42	0.42	15001
2	0.35	0.26	0.30	11219
3	0.62	0.55	0.58	9191
accuracy			0.47	52940
macro avg	0.47	0.46	0.46	52940
weighted avg	0.46	0.47	0.46	52940

4. ROC CURVE



7.7 GRADIENT BOOSTING CLASSIFIER

Gradient Boosting works by iteratively adding new decision trees to the ensemble, where each new tree tries to correct the errors made by the previous trees. It uses a gradient descent algorithm to optimize the loss function, and it can handle both regression and classification problems

Accuracy Score:

The accuracy score is calculated with a calculated optimal threshold

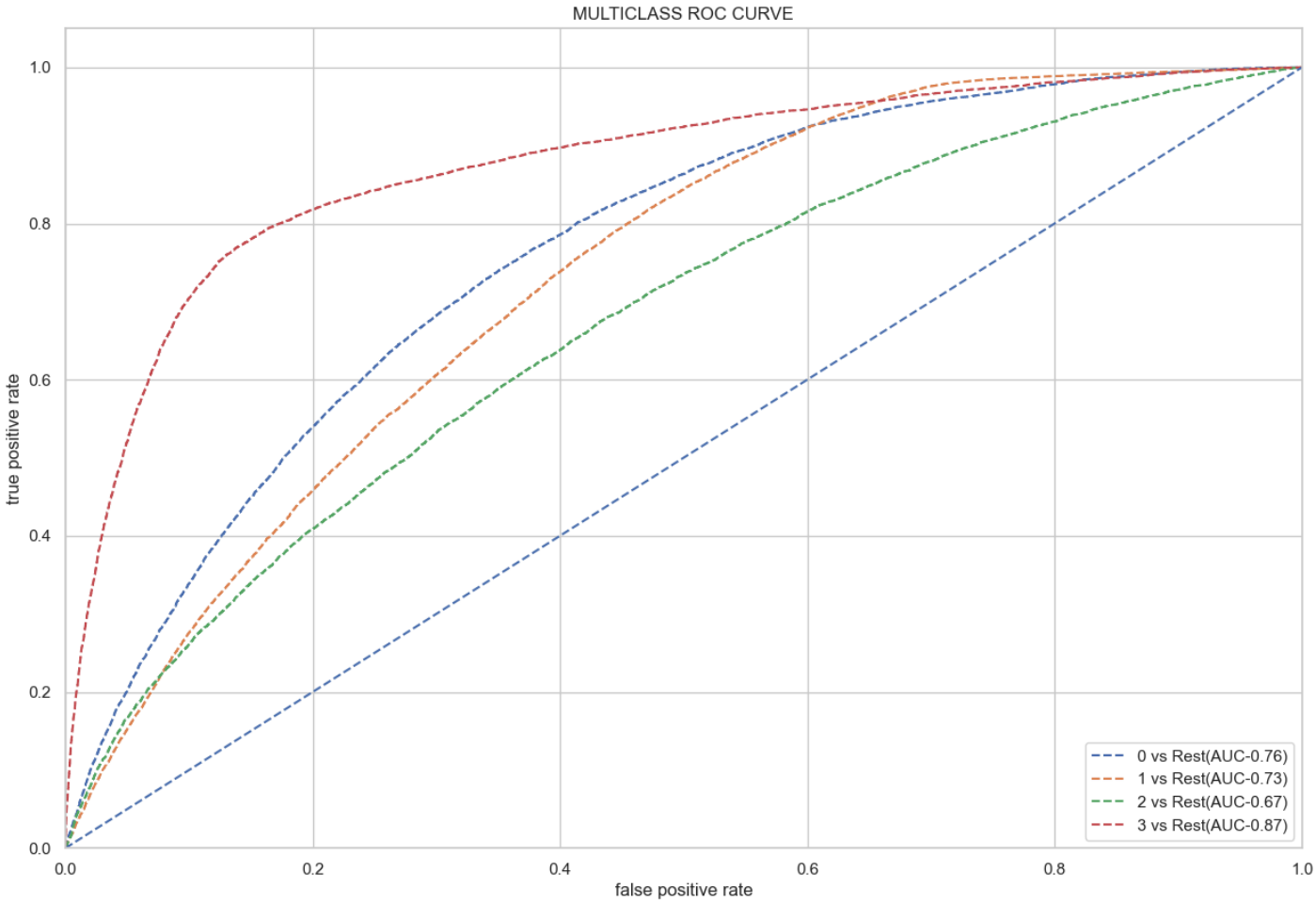
```
Accuracy score:  0.5149981110691348
```

2. Confusion matrix

3. Classification Report:

classification report:	precision	recall	f1-score	support
0	0.52	0.71	0.60	17529
1	0.46	0.51	0.48	15001
2	0.49	0.08	0.13	11219
3	0.61	0.69	0.65	9191
accuracy			0.51	52940
macro avg	0.52	0.50	0.47	52940
weighted avg	0.51	0.51	0.48	52940

4. ROC CURVE


MULTICLASS ROC CURVE
true positive rate
false positive rate
0.0
0.2
0.4
0.6
0.8
1.0
0 vs Rest(AUC-0.76)
1 vs Rest(AUC-0.73)
2 vs Rest(AUC-0.67)
3 vs Rest(AUC-0.87)

7.8. XGBOOST CLASSIFIER:

XGBoost is an advanced implementation of gradient boosting that uses a more efficient gradient boosting algorithm, regularized learning, and tree pruning to reduce overfitting and improve the speed and accuracy of the model.

Train Dataset Evaluation:

1. **Accuracy Score:**

```
Accuracy score:  0.523290517567057
```

2. **Confusion matrix**

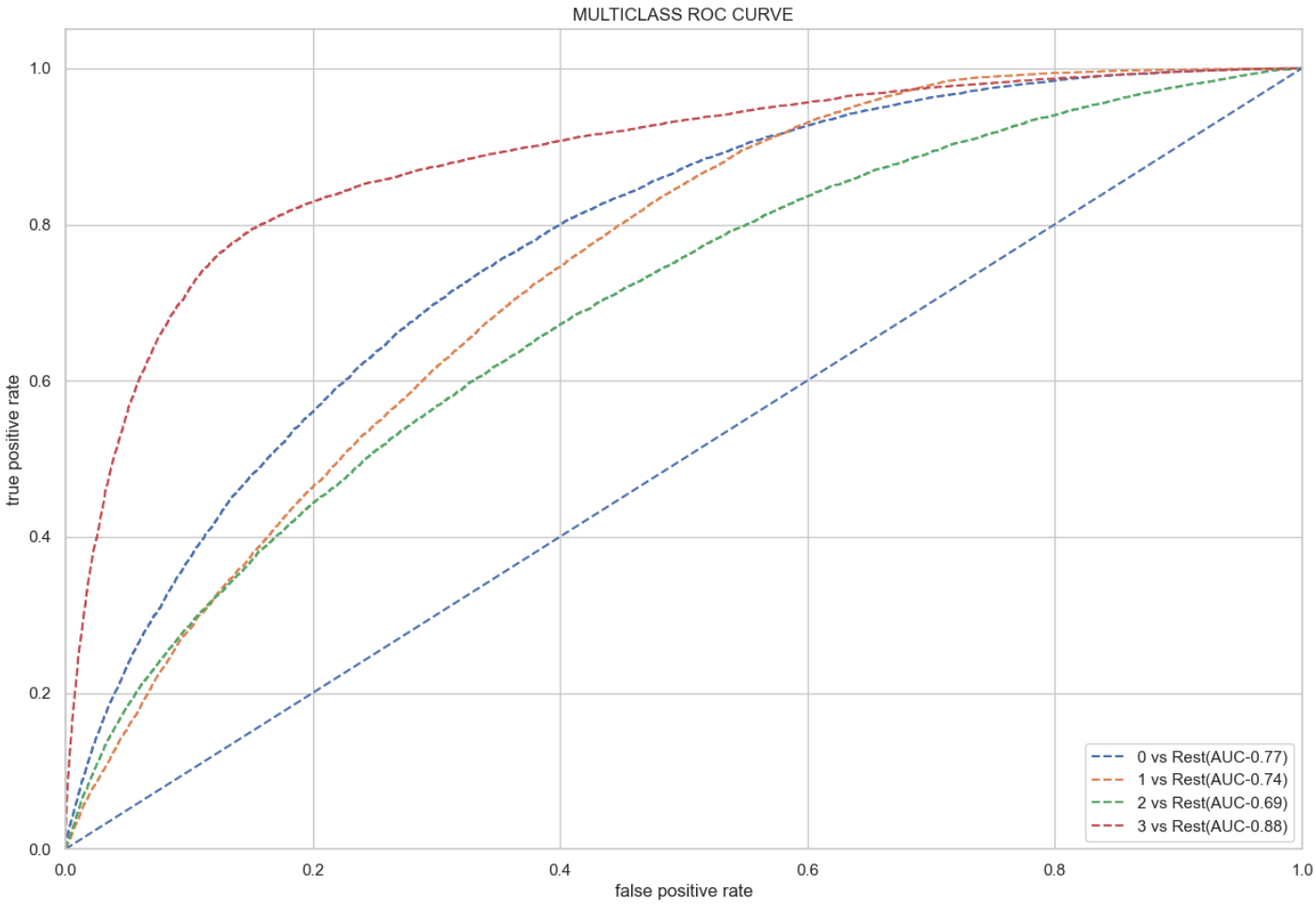
3. Classification Report:

```
classification report:
              precision    recall  f1-score   support

           0       0.53      0.68      0.60     17529
           1       0.46      0.54      0.49     15001
           2       0.47      0.17      0.25     11219
           3       0.65      0.65      0.65      9191

    accuracy                           0.52     52940
   macro avg       0.53      0.51      0.50     52940
weighted avg       0.52      0.52      0.50     52940
```

4. ROC CURVE



7.9. LIGHT GBM CLASSIFIER

Light GBM (Light Gradient Boosting Machine) is a gradient boosting framework that uses a highly optimized decision tree algorithm to improve the speed and accuracy of predictions. It uses a gradient-based one-side sampling (GOSS) technique and exclusive feature bundling (EFB) to reduce overfitting and improve the efficiency of the learning process.

1. **Accuracy Score:**

```
Accuracy score:  0.523290517567057
```

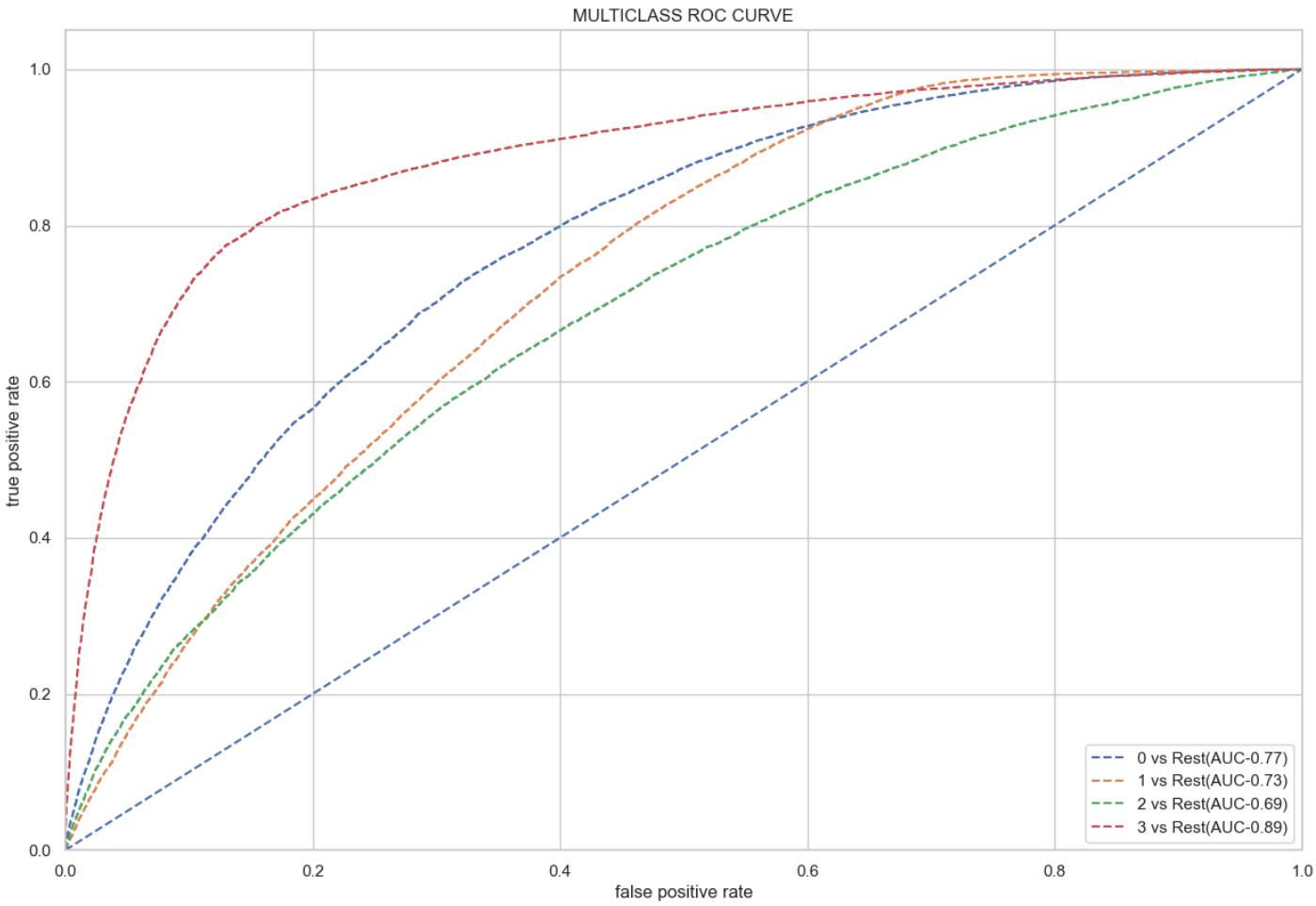
2. **Confusion matrix:**

3. **Classification Report:**

classification report:

	precision	recall	f1-score	support
0	0.53	0.68	0.60	17529
1	0.46	0.54	0.49	15001
2	0.47	0.17	0.25	11219
3	0.65	0.65	0.65	9191
accuracy			0.52	52940
macro avg	0.53	0.51	0.50	52940
weighted avg	0.52	0.52	0.50	52940

4. ROC CURVE



7.11. HYPER-PARAMETER TUNNING

Hyperparameter tuning is the process of selecting the optimal values of hyperparameters that control the behavior of a machine learning algorithm. Hyperparameters are parameters that are not learned during training but are set before the training process, such as the learning rate, regularization strength, number of hidden layers

After tuning the XGBoost Classifier using GridSearchCV , there were good changes in the accuracy and F1 score. They were higher than the base and other models.

```
Accuracy score:  0.522100491122025
```

The conclusion is that XGBoost Classifier is performing well in our scenario and it is best to follow other models for better accuracy score.

7.12. LGBM CLASSIFIER

LGBM Classifier is a machine learning algorithm used for classification tasks that utilizes the Light Gradient Boosting Machine (LightGBM) framework. It is a gradient boosting algorithm that uses decision trees as base models, and it is designed to be fast, efficient, and scalable for large datasets.

Train Dataset Evaluation:

Accuracy Score:

The accuracy score is calculated with a calculated optimal threshold

```
Training Accuracy
0.508004420139971
Testing Accuracy
0.5039289761994711
```

2. Classification Report:

	precision	recall	f1-score	support
0	0.55	0.60	0.57	17529
1	0.44	0.57	0.50	15001
2	0.32	0.09	0.14	11219
3	0.59	0.72	0.65	9191
accuracy			0.50	52940
macro avg	0.47	0.49	0.46	52940
weighted avg	0.48	0.50	0.47	52940

8. UNDER SAMPLING TECHNIQUE:

Under sampling Technique for balancing the Imbalanced data.

Metrics	Logistic Regression - Whole data	Logistic Regression - Under-sampled data
Accuracy	0.49	0.26
Precision	0.50	0.31
Recall	0.68	0.59
F1-Score	0.58	0.39

8.1 DECISION TREE CLASSIFIER

BUILDING MODEL BASED ON DECISION TREE CLASSIFIER AND EVALUATION

Train Dataset Evaluation:

1.Accuracy Score:

The accuracy score is calculated with a calculated optimal threshold.

```
Accuracy score:  0.4162754303599374
```

2. Confusion matrix:

3.Classification report:

	precision	recall	f1-score	support
0	0.45	0.64	0.53	866
1	0.34	0.31	0.32	856
2	0.26	0.09	0.14	854
3	0.25	0.11	0.16	822
4	0.49	0.81	0.61	1075
accuracy			0.42	4473
macro avg	0.36	0.39	0.35	4473
weighted avg	0.36	0.42	0.36	4473

8.2. RANDOM FOREST CLASSIFIER MODEL

Train Dataset Evaluation:

1.Accuracy Score:

The accuracy score is calculated with a default threshold of 0.5 .

```
Accuracy score:  0.3456293315448245
```

3. Classification Report:

```
classification report:
              precision    recall  f1-score   support

           0       0.42      0.42      0.42       866
           1       0.28      0.28      0.28       856
           2       0.23      0.22      0.22       854
           3       0.24      0.24      0.24       822
           4       0.50      0.52      0.51      1075

    accuracy                           0.35      4473
   macro avg       0.33      0.34      0.33      4473
weighted avg       0.34      0.35      0.34      4473
```

8.3. ADA BOOST CLASSIFIER

BUILDING MODEL BASED ON ADA BOOST CLASSIFIER AND EVALUATION

Train Dataset Evaluation:

1. Accuracy Score:

The accuracy score is calculated with a calculated optimal threshold

```
Accuracy score:  0.4164989939637827
```

2. Classification Report:

Classification report:

	precision	recall	f1-score	support
0	0.47	0.56	0.51	866
1	0.34	0.32	0.33	856
2	0.29	0.21	0.24	854
3	0.28	0.20	0.23	822
4	0.53	0.71	0.61	1075
accuracy			0.42	4473
macro avg	0.38	0.40	0.38	4473
weighted avg	0.39	0.42	0.40	4473

8.4. BAGGING CLASSIFIER

BUILDING MODEL BASED ON BAGGING CLASSIFIER AND EVALUATION

Train Dataset Evaluation:

1. Accuracy Score:

The accuracy score is calculated with a calculated optimal threshold

```
Accuracy score:  0.41582830315224684
```

2. Classification Report:

```
classification report:
              precision    recall  f1-score   support

           0       0.44      0.63      0.52       866
           1       0.35      0.32      0.33       856
           2       0.25      0.12      0.16       854
           3       0.29      0.19      0.23       822
           4       0.52      0.73      0.61      1075

    accuracy                           0.42      4473
   macro avg       0.37      0.40      0.37      4473
weighted avg       0.38      0.42      0.38      4473
```

8.5 GRADIENT BOOSTING CLASSIFIER

BUILDING MODEL BASED ON GRADIENT BOOSTING CLASSIFIER AND EVALUATION

Train Dataset Evaluation:

1. **Accuracy Score:**

The accuracy score is calculated with a calculated optimal threshold

```
Accuracy score:  0.39145987033310975
```

2. Classification Report:

Classification report:

	precision	recall	f1-score	support
0	0.44	0.57	0.50	866
1	0.31	0.31	0.31	856
2	0.25	0.22	0.23	854
3	0.27	0.22	0.24	822
4	0.57	0.58	0.57	1075
accuracy			0.39	4473
macro avg	0.37	0.38	0.37	4473
weighted avg	0.38	0.39	0.38	4473

8.6 XGBOOST CLASSIFIER

BUILDING MODEL BASED ON XGBOOSTING CLASSIFIER AND EVALUATION

Train Dataset Evaluation:

Accuracy Score:

The accuracy score is calculated with a calculated optimal threshold

```
Accuracy score:  0.43639615470601384
```

2. Classification Report:

	precision	recall	f1-score	support
0	0.47	0.70	0.56	866
1	0.36	0.38	0.37	856
2	0.28	0.12	0.16	854
3	0.30	0.15	0.20	822
4	0.53	0.74	0.62	1075
accuracy			0.44	4473
macro avg	0.39	0.42	0.38	4473
weighted avg	0.39	0.44	0.40	4473

8.7 LIGHT GBM CLASSIFIER

BUILDING MODEL BASED ON LIGHT GBM CLASSIFIER AND EVALUATION

Train Dataset Evaluation:

Accuracy Score:

The accuracy score is calculated with a calculated optimal threshold

```
Accuracy score:  0.42946568298680976
```

2. Classification Report:

classification report:

	precision	recall	f1-score	support
0	0.50	0.64	0.56	866
1	0.34	0.36	0.35	856
2	0.28	0.17	0.21	854
3	0.29	0.20	0.24	822
4	0.55	0.69	0.62	1075
accuracy			0.43	4473
macro avg	0.39	0.41	0.40	4473
weighted avg	0.40	0.43	0.41	4473

8.8 HYPER-PARAMETER TUNNING

After tuning the Random Forest Classifier using GridSearchCV , there were good changes in the accuracy and F1 score. They were higher than the base and other models.

Accuracy score: 0.42946568298680976

The conclusion is that Random Forest Classifier is performing well in our scenario and it is best to follow other models for better accuracy score.

After tuning the XGBoost classifier using GridSearchCV , there were good changes in the accuracy and F1 score. They were higher than the base and other models.

```
Accuracy score:  0.4214173932483792
```

The conclusion is that XGBoost is performing well in our scenario and it is best to follow other models for better accuracy score.

8.9 Best Model:

Whole Data-set:

MODEL	ACCURACY
Logistic Regression	0.495
Decision Tree	0.412
Random Forest	0.480
XGBoost	0.523
Light GBM	0.523
Tuned XGBoost	0.522

Under-sampled Data-set:

MODEL	ACCURACY
Logistic Regression	0.416
Decision Tree	0.345
Random Forest	0.416
XGBoost	0.429
Light GBM	0.429
Tuned XGBoost	0.522

9.BUSINESS SUGGESTIONS

Since we are dealing with a multi-class classification problem, we can consider the f1 score for each of the classes since the cost of both false positive and false negative is high.

9.1 .PROJECT OUTCOME

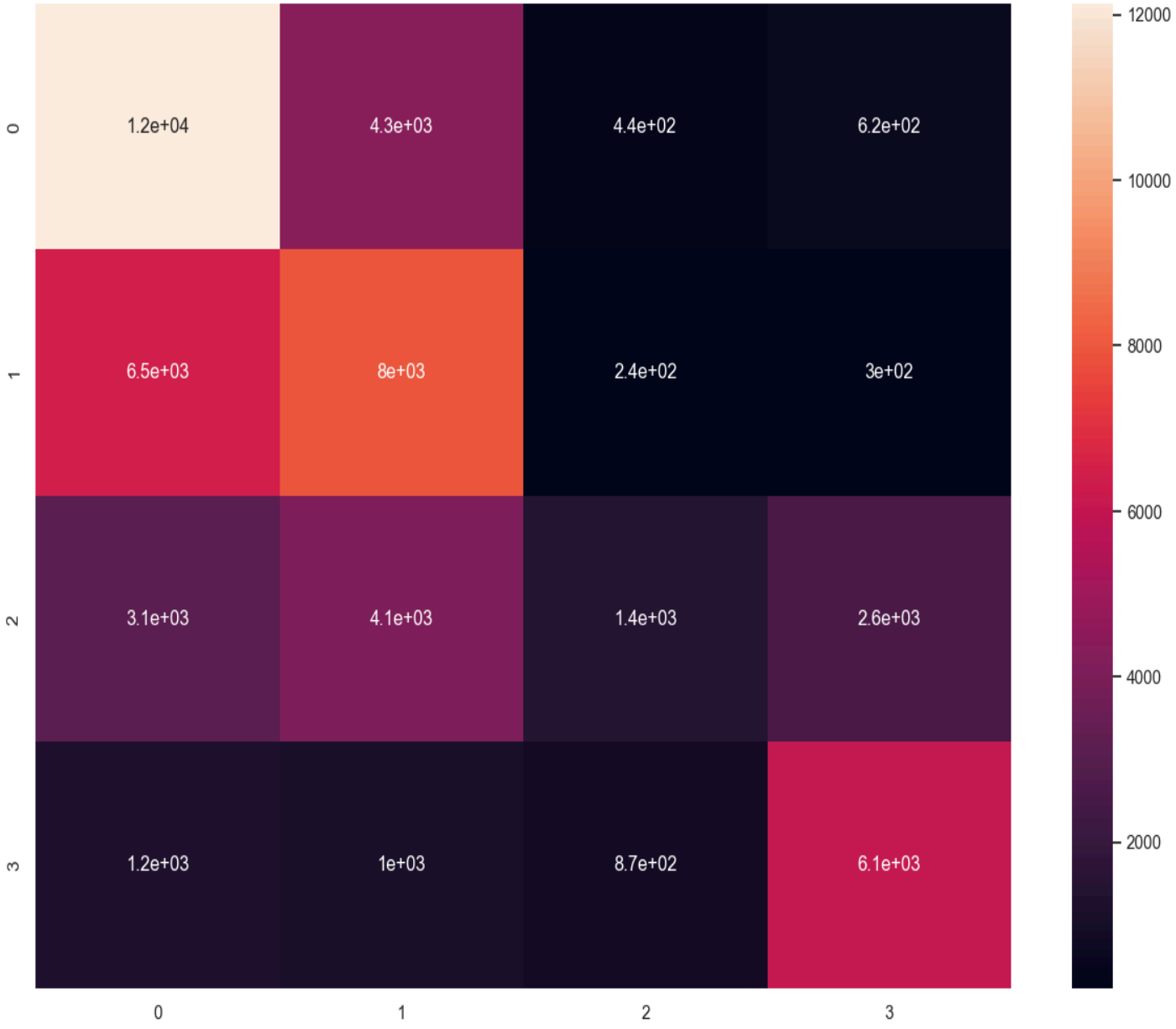
- For our project we have decided that the best suite model was XGBoost. This model has produced the results with the highest level of accuracy.

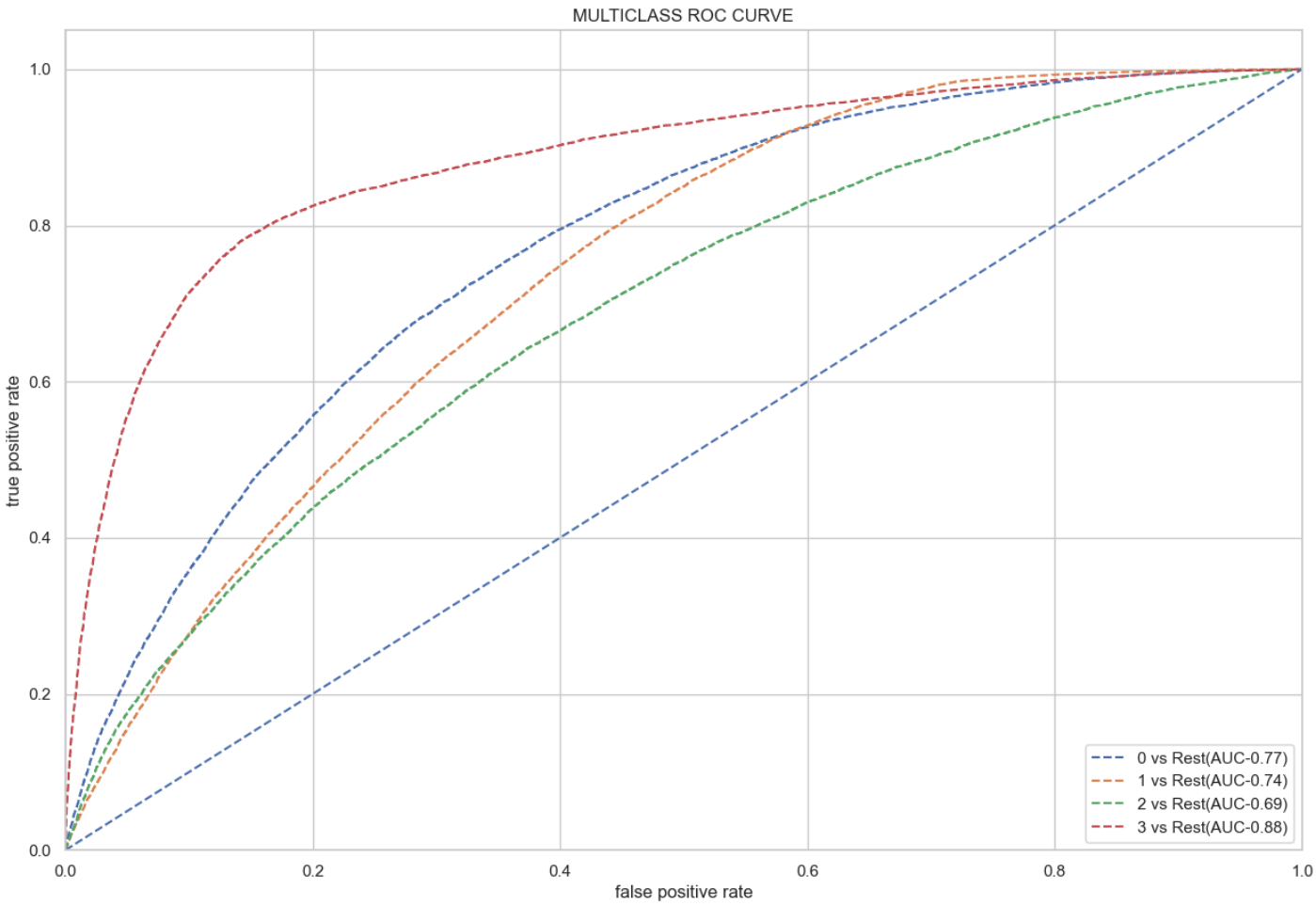
Classification report:

classification report:

	precision	recall	f1-score	support
0	0.53	0.68	0.60	17529
1	0.46	0.54	0.49	15001
2	0.47	0.17	0.25	11219
3	0.65	0.65	0.65	9191
accuracy			0.52	52940
macro avg	0.53	0.51	0.50	52940
weighted avg	0.52	0.52	0.50	52940

Confusion matrix:




MULTICLASS ROC CURVE
true positive rate
false positive rate
0.0
0.2
0.4
0.6
0.8
1.0
0 vs Rest(AUC-0.77)
1 vs Rest(AUC-0.74)
2 vs Rest(AUC-0.69)
3 vs Rest(AUC-0.88)

10. REFERENCES

1. COVID-19 Hospitals Treatment Plan | Kaggle
2. Stasi, C., Fallani, S., Voller, F., & Silvestri, C. (2020). Treatment for COVID-19: An overview. European journal of pharmacology, 889, 173644.
3. Felsenstein, S., Herbert, J. A., McNamara, P. S., & Hedrich, C. M. (2020). COVID-19: Immunology and treatment options. Clinical immunology, 215, 108448.
4. Waller, M. A., & Fawcett, S. E. (2013). Data science, predictive analytics, and big data: a revolution that will transform supply chain design and management. Journal of Business Logistics, 34(2), 77-84.
5. Raschka, S., Patterson, J., & Nolet, C. (2020). Machine learning in python: Main developments and technology trends in data science, machine learning, and artificial intelligence. Information, 11(4), 193.