FINAL PROJECT PRESENTATION

Readmission Of Patients

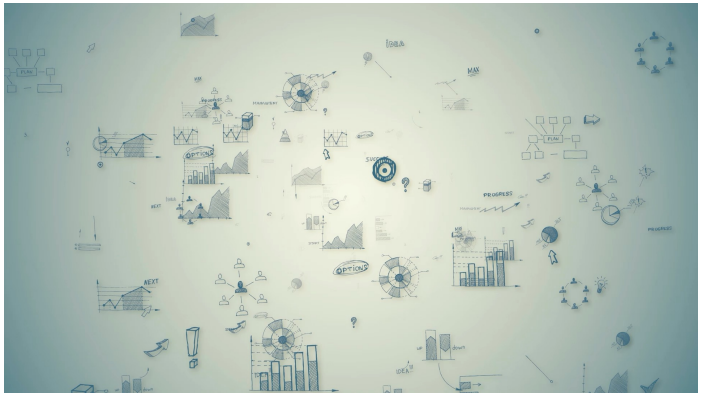
USE CASE OF THE PROJECT

- For the course project, we will be investigating the problem of predicting readmission of hospital patients. Readmission of patients following discharge from hospital places an enormous and very expensive burden on the US healthcare system (estimated at $40B in 2011).
- The objective of this project is to predict the early (<30 days) readmission given the patient's features. We are trying to predict if a patient gets admitted before 30 days or not , so that the allocation can be better managed.

THE DATASET

- We will use this UCI dataset. The dataset represents 10 years (1999-2008) of clinical care at 130 US hospitals and integrated delivery networks. It includes over 50 features representing patient and hospital outcomes. Information was extracted from the database for encounters that satisfied the following criteria.
  - It is an inpatient encounter (a hospital admission).
  - It is a diabetic encounter, that is, one during which any kind of diabetes was entered into the system as a diagnosis.
  - The length of stay was at least 1 day and at most 14 days.
  - Laboratory tests were performed during the encounter.
  - Medications were administered during the encounter.
- The data contains such attributes as patient number, race, gender, age, admission type, time in hospital, medical specialty of admitting physician, number of lab test performed, HbA1c test result, diagnosis, number of medications, diabetic medications, number of outpatient, inpatient, and emergency visits in the year before the hospitalization, etc.

EXPLORING THE DATA

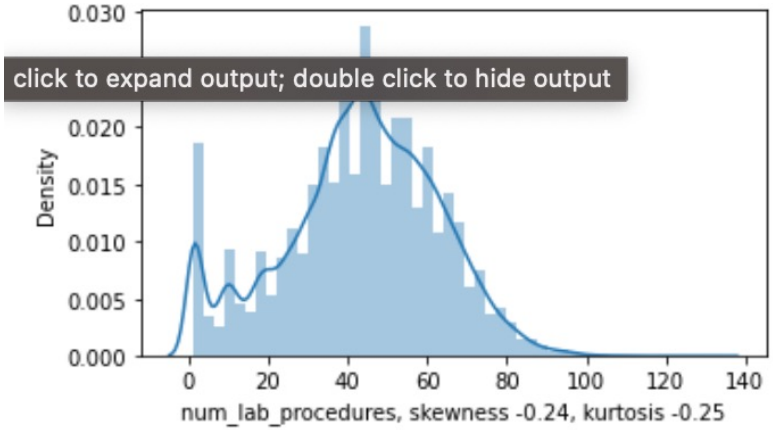


- Basic Data Information:
  - The Train Dataset contains 76324 rows while the Test Dataset contains 25442 rows.
  - Number of Feature columns: 50
  - Number of Target column = 1
  - Number of numeric columns: 13
  - Number of categorical columns: 37
- Target Column:
  - We converted the target column from multivariate to Bivariate by grouping category '*No*' and category '*>30*' together as "other" and '*<30*' was considered as a seperate category.
  - The number of readmitted cases before 30 days were 11357 whereas the number of not admitted or readmitted after 30 days were 90409.

UNIVARIATE ANALYSIS OF NUMERIC COLUMNS

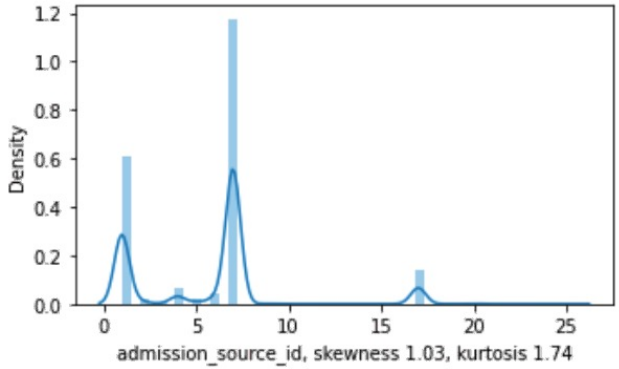
We performed Univariate analysis on Categorical as well as numeric columns. Here are some of the insights:

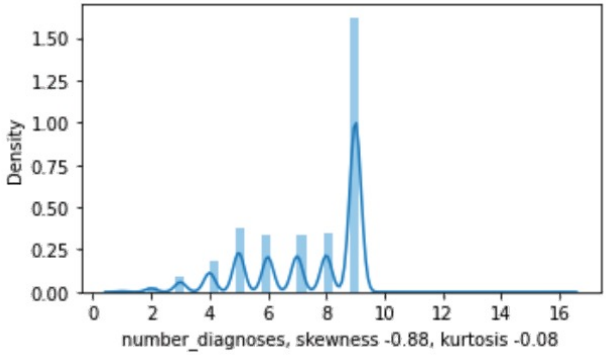
1. num_lab procedures distribution is fairly symmetrical.



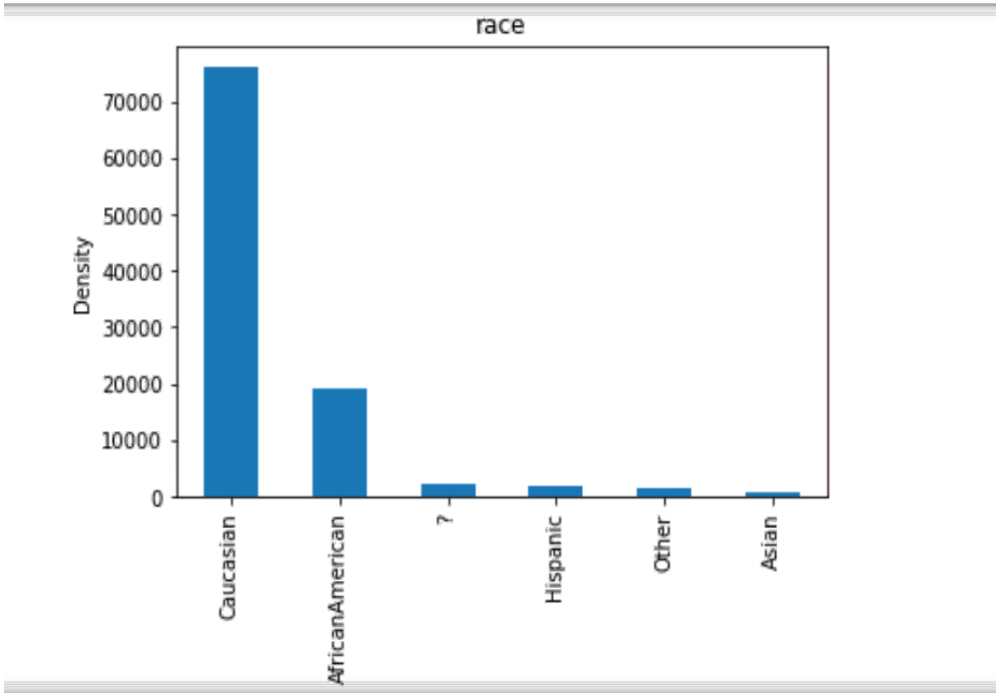
Univariate analysis

- admission_source_id, number_diagnosis columns are moderaltely skewed.

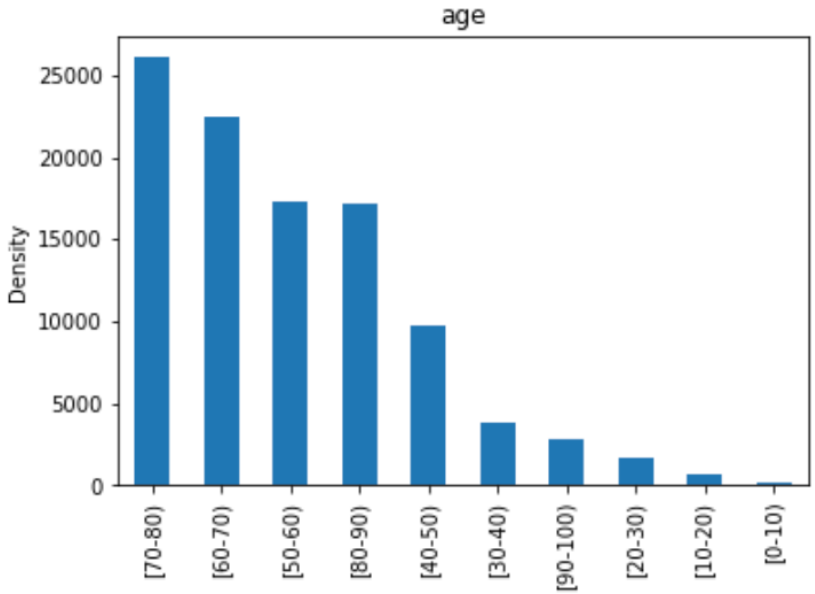




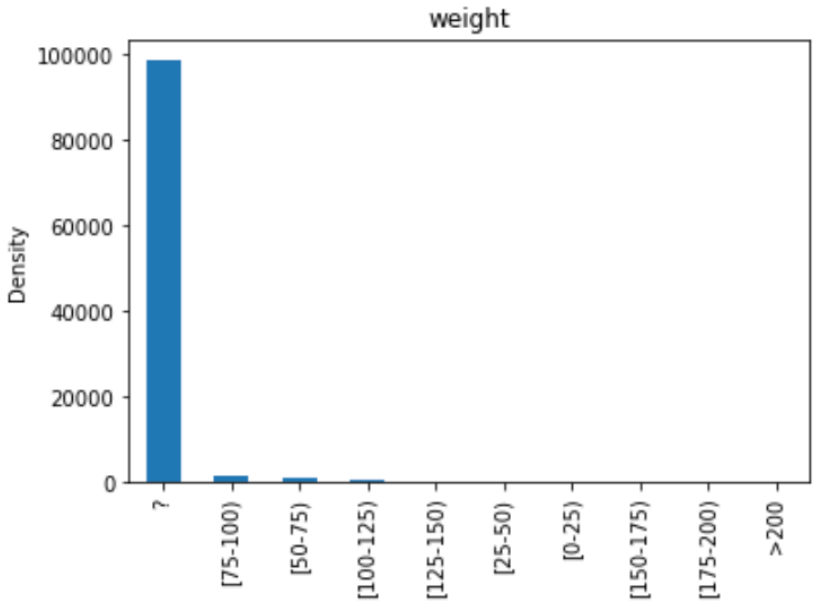
- Race columns has 5 races and also some null values represented by ?, The data has most entries of people who are originated from caucasian or are African American.



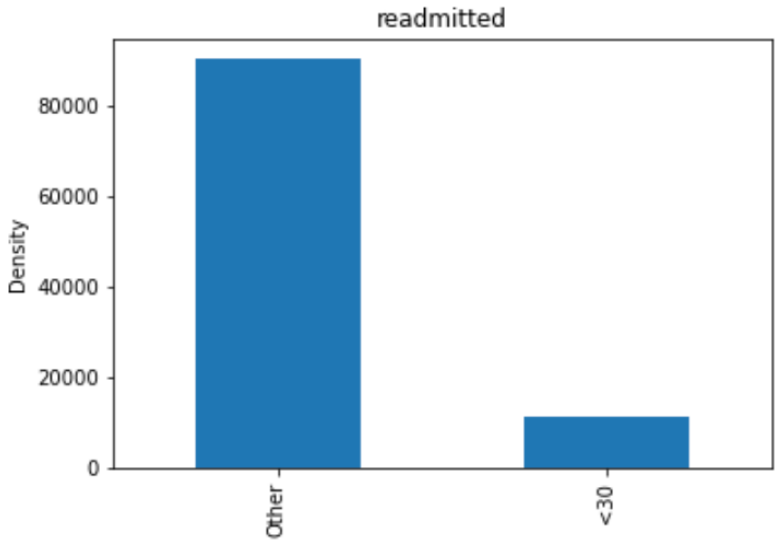
- Age 50-70 as expected has more enteries as they have more chances to get admitted in a hospital.



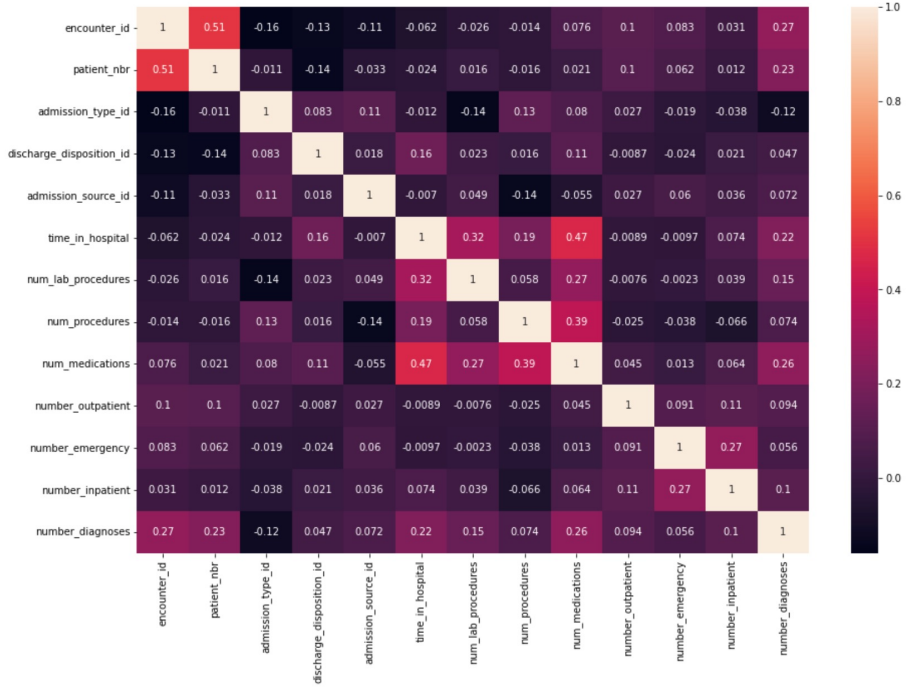
- 97% values in weight are missing so it will be good to drop this column.



Target column is highly imbalanced.



Bivariate analysis

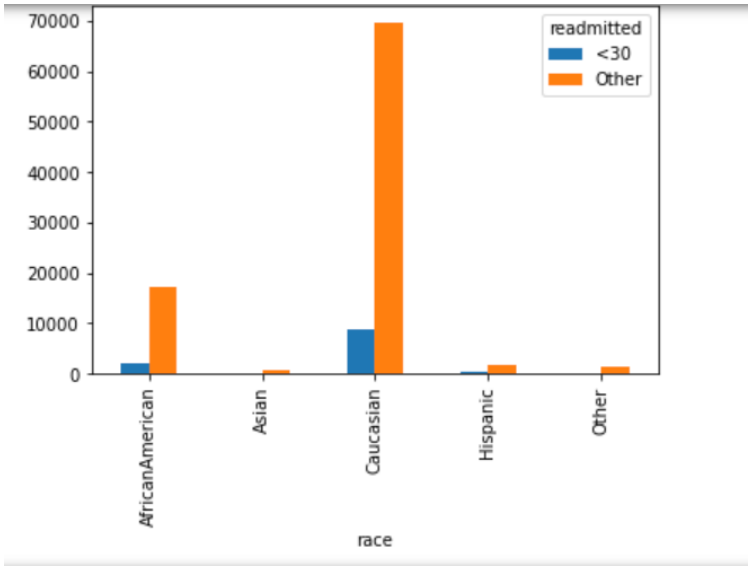


Insights from above heamap:

- encounter_id and patient_nbr have high positive correlation
- time_in_hospital and num_medication also have high positive correlation
- time_in_hospital, num_lab_procedures, num_procedures and num_medication also have good positive correlation

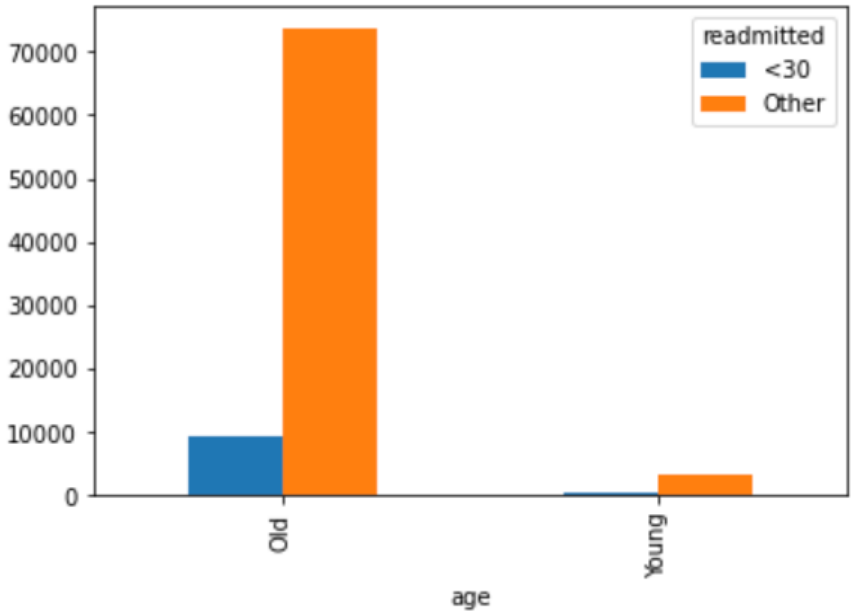
TARGET COLUMN VS CATEGORICAL COLUMNS

- Race which belongs to Caucasian and African American have more chances of readmitted within 30 days



TARGET COLUMN VS CATEGORICAL COLUMNS

- Age 50-90 have more chances of readmitting within 30 days.



SANITIZING THE DATA

Columns Dropped:

- **Weight** column had 97% missing values so dropping that column
- Dropping **examide** and **citoglipton** since they had only one category in them.
- Dropped columns based on low variance.
- Dropped **encounter_id** and **patient_nbr** because they are just all unique IDs.
- Dropping ***medical_speciality*** as its contain lots of categories in it.

DATA IMPUTATION

Imputed Null (?) of following columns with mode:

- *race*
- *payer_code*
- *medical_specialty*
- *diag3*

DATA BINNING

Binned Diag1, Diag2 , Diag3 as given in ICD-9 code:

- 001–139: infectious and parasitic diseases
- 140–239: neoplasms
- 240–279: endocrine, nutritional and metabolic diseases, and immunity disorders
- 280–289: diseases of the blood and blood-forming organs
- 290–319: mental disorders
- 320–389: diseases of the nervous system and sense organs
- 390–459: diseases of the circulatory system
- 460–519: diseases of the respiratory system
- 520–579: diseases of the digestive system
- 580–629: diseases of the genitourinary system
- 630–679: complications of pregnancy, childbirth, and the puerperium
- 680–709: diseases of the skin and subcutaneous tissue
- 710–739: diseases of the musculoskeletal system and connective tissue
- 740–759: congenital anomalies
- 780–799: symptoms, signs, and ill-defined conditions
- 800–999: injury and poisoning
- 760–779: certain conditions originating in the perinatal period
- E and V codes: external causes of injury and supplemental classification

BINNING AGE

BINNING AGE INTO YOUNG, MIDDLE-AGE AND OLD

- age greater than or equal to 20: Young
- age greater 20 and smaller than or equal to 50: Middle Age
- age greater than 50: Old

PREPROCESSING THE DATA

- *Encoded Categorical columns*
  - Example:
    - Race column before encoding:
      - [Screenshot]
    - Race column after encoding
      - [screenshot]
- *Scaled the data using standard scaler.*
  - Explanation of standard scaler
    - Standardize features by removing the mean and scaling to unit variance.The standard score of a sample x is calculated as z = (x - u) / s. where u is the mean of the training samples or zero if with_mean=False, and s is the standard deviation of the training samples or one if with_std=False.
  - Data before scaling
    - []screenshot]
  - Data after scaling
    - [screenshot]
- *Encoding target column*
  - Other class was converted to zero and <30 was converted to 1
  - Before and after screenshots

FEATURE SELECTION

Performed PCA with different variation of resulting number of features

Before and after data shape and some explanation of how you did it.

Write like we used PCA from sklearn

Some information about the method from sklearn: Linear dimensionality reduction using Singular Value Decomposition of the data to project it to a lower dimensional space. The input data is centered but not scaled for each feature before applying the SVD.

MODEL TRAINING AND EVALUATION

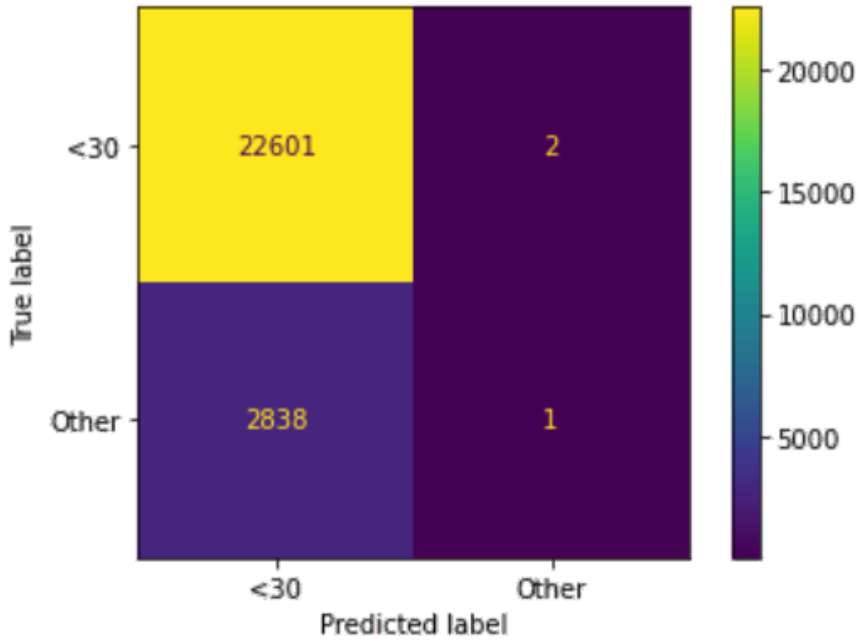
Tried the following classification models:

1. Logistic Regression
2. Random Forest
3. Decision Tree Classifier
4. Adaboost Classifier
5. XGBoost

LOGISTIC REGRESSION

Accuracy: 88.84% .

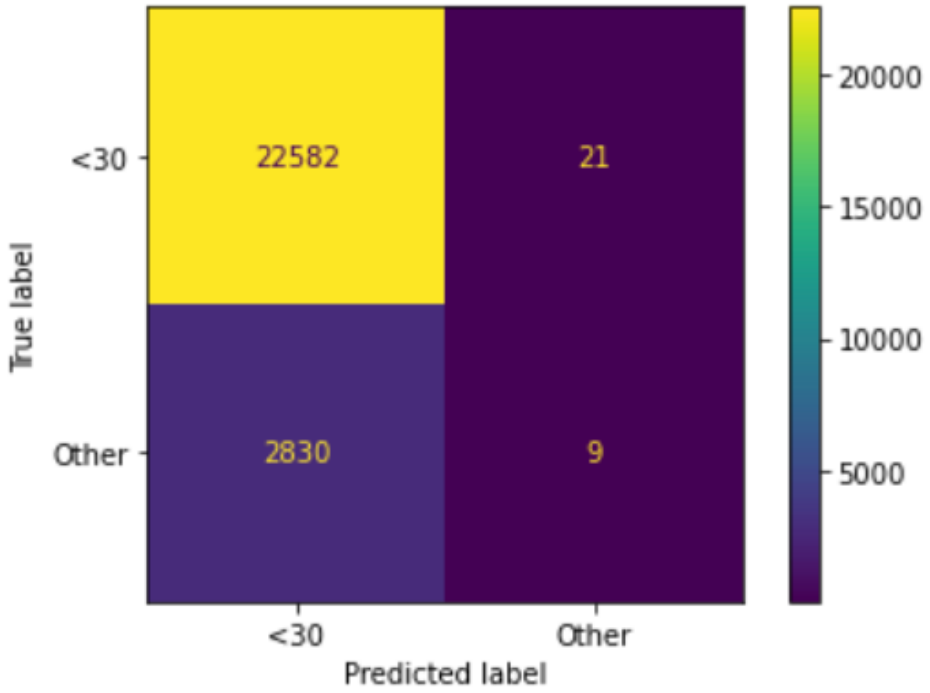
Confusion Matrix:



RANDOM FOREST

Accuracy: 88.79%

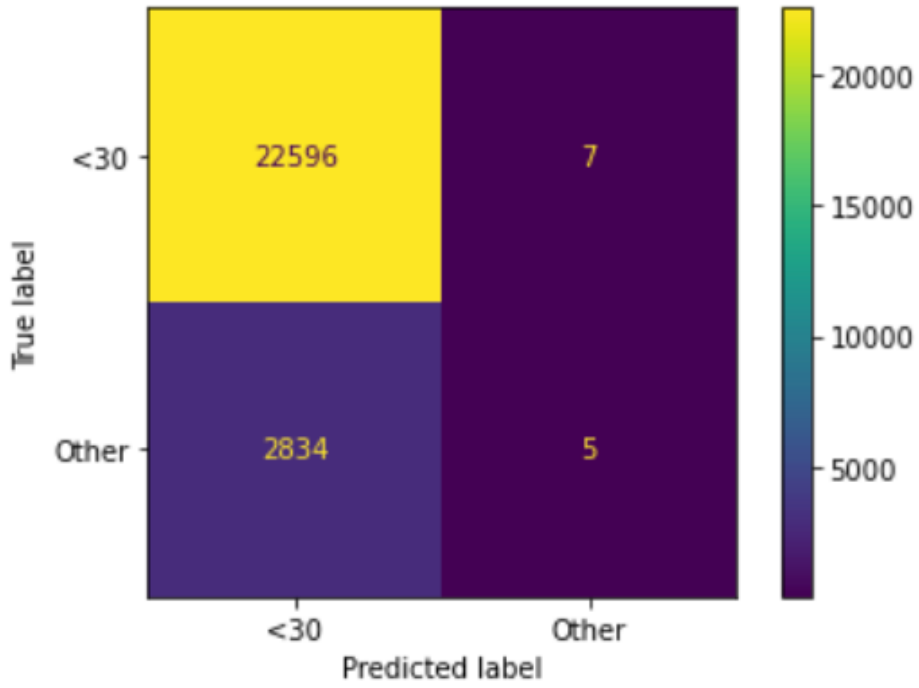
Confusion Matrix:



DECISION TREE CLASSIFIER

Accuracy of Decision Tree Model: 88.83%

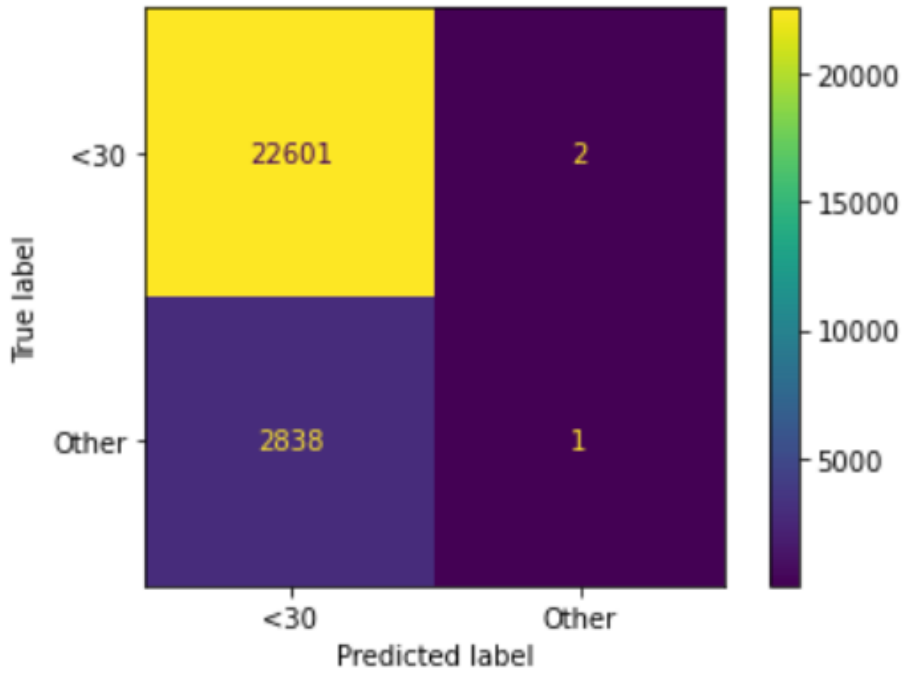
Confusion Matrix:



ADABOOST CLASSIFIER

Accuracy of Adaboost Model: 88.84%

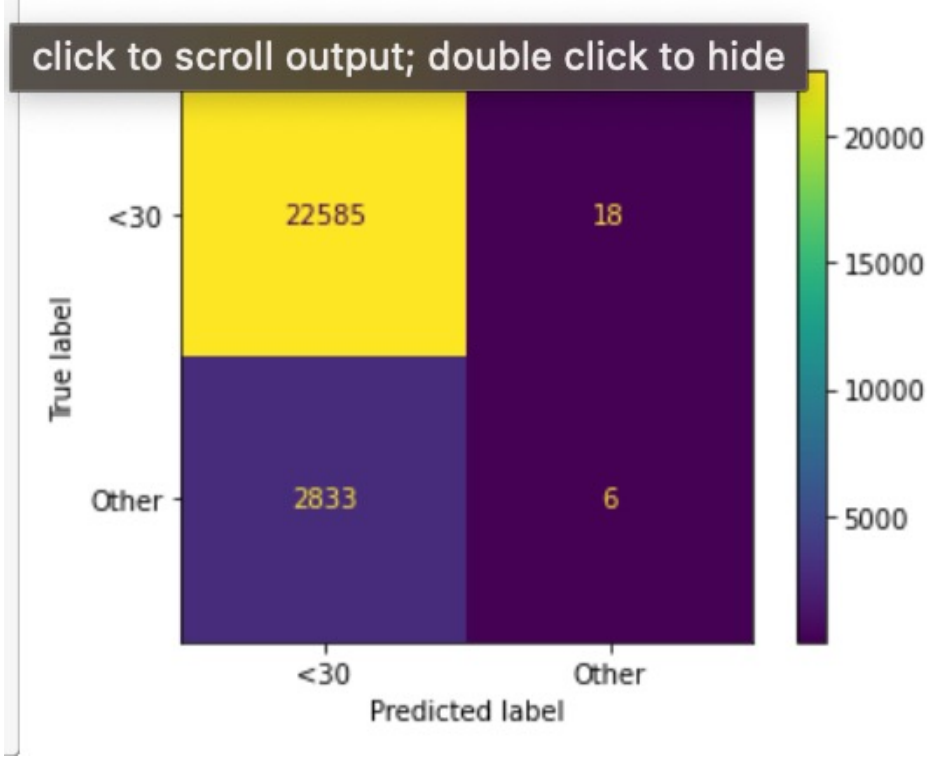
Confusion Matrix:



XGBOOST

Accuracy of XGBoost Model: 88.79%

Confusion Matrix:



HYPER-PARAMETER TUNING

- Performed hyperparameter tuning over Random Forest.
- Parameters used by Random Forest in default setting are :

```
{'bootstrap': True,
 'ccp_alpha': 0.0,
 'class_weight': None,
 'criterion': 'gini',
 'max_depth': None,
 'max_features': 'auto',
 'max_leaf_nodes': None,
 'max_samples': None,
 'min_impurity_decrease': 0.0,
 'min_samples_leaf': 1,
 'min_samples_split': 2,
 'min_weight_fraction_leaf': 0.0,
 'n_estimators': 100,
 'n_jobs': None,
 'oob_score': False,
 'random_state': None,
 'verbose': 0,
 'warm_start': False}
```

- USed RandomizedSeachCV from Sklearn to search for best hyperparameters.
  - Following hyperparameter grid was used:
    - {'n_estimators': [100, 150, 200],

      'max_features': ['auto', 'sqrt'],

      'max_depth': [10, 20, 30, 40, None],

      'min_samples_split': [2, 5, 10],

      'min_samples_leaf': [1, 2, 4],

      'bootstrap': [True, False]}

- Performed Random search of parameters, using K fold cross validation
- Predicted with the best fit model.
- Results:

Accuracy of resulting Random Forest Model: 88.84%

Confusion Matrix: