Neshan data science boot camp

Task 3 report

Sara Hadavi

Goal of project:

We are supposed to answer a question about whether we could predict the ICU admission of a COVID-19 patient based on its medical features or not.

Steps:

At first we proceed by data cleaning and preprocessing steps(EDA):

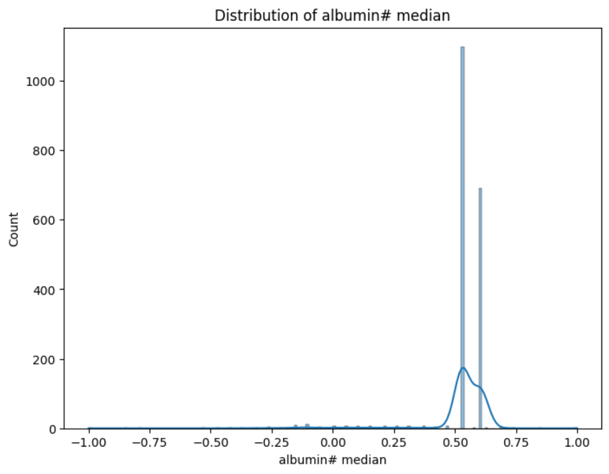
- delete records(rows) which has a value other than 1 or 0 in their target (last) column from dataset.
- delete the rows which have missing value in target column.
- Data cleaning:
    - detect columns which are of categorical type for further encoding. We know that there are mistaken entries in some columns such that a column which is in fact numerical, could contain string values or other objects. Identify numerical columns if more than 90% of their values are of integer or float type.
    - 3 categorical columns were found:
        - `['age', 'observation window', 'data tags']`
        - now, this 3 column may also contain mistaken values such as integers or floats or even string objects which are of very low frequency in entire column. We first go through each of these columns, replace the mistaken or irrelevant items(values which are of frequency less than twice) with most common value in that column. then fill all the missing values with the most frequent element in that column.
        - Then we make sure there is no more missing value or value of wrong datatype present in this columns. Now, we encode this 3 categorical columns using pandas LabelEncoder method.
        - Encode categorical columns using LabelEncoder and save mappings:
            - Column: age
            - Missing values: 0
            - Data type: int64
            - Original to Encoded mapping: {'10th': 0, '20th': 1, '30th': 2, '40th': 3, '50th': 4, '60th': 5, '70th': 6, '80th': 7, '90th': 8, 'Above 90th': 9, 'more than 40': 10}
            - --------------------
            - Column: observation window
            - Missing values: 0
            - Data type: int64
            - Original to Encoded mapping: {'0-2': 0, '2-4': 1, '4-6': 2, '6-12': 3, 'ABOVE_12': 4}

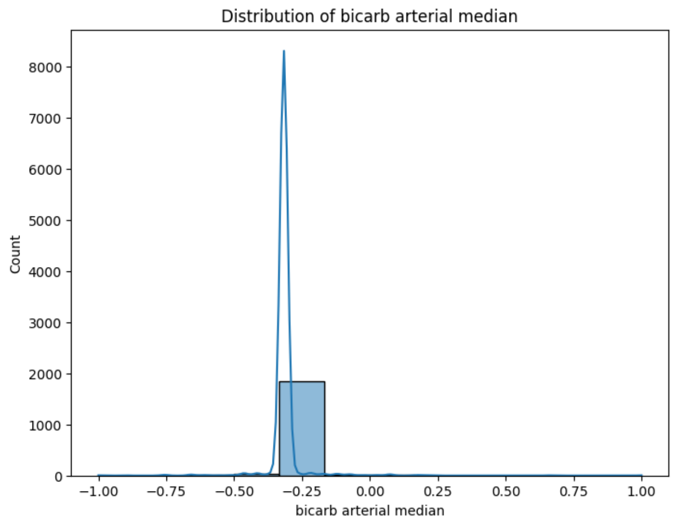
- --------------------
- Column: data tags
- Missing values: 0
- Data type: int64
- Original to Encoded mapping: {'Kidney disease': 0, 'Lung cancer': 1, 'Lung cancer, Motor Neurone Disease': 2, 'Lung cancer, Smoker': 3, 'Lung cancer, asthma': 4, 'Lung cancer, asthma, Kidney disease, Motor Neurone Disease, Smoker': 5, 'Lung cancer, asthma, Motor Neurone Disease': 6, 'Lung cancer, asthma, Motor Neurone Disease, Smoker': 7, 'Lung cancer, asthma, Smoker': 8, 'Lung cancer, heart disease': 9, 'Lung cancer, heart disease, Motor Neurone Disease': 10, 'Lung cancer, heart disease, asthma, Motor Neurone Disease': 11, 'Motor Neurone Disease': 12, 'Motor Neurone Disease, Smoker': 13, 'Nothing': 14, 'Smoker': 15, 'Unknown': 16, 'asthma': 17, 'asthma, Kidney disease': 18, 'asthma, Kidney disease, Motor Neurone Disease': 19, 'asthma, Kidney disease, Smoker': 20, 'asthma, Motor Neurone Disease': 21, 'asthma, Motor Neurone Disease, Smoker': 22, 'asthma, Smoker': 23, 'heart disease': 24, 'heart disease, Kidney disease': 25}
- --------------------

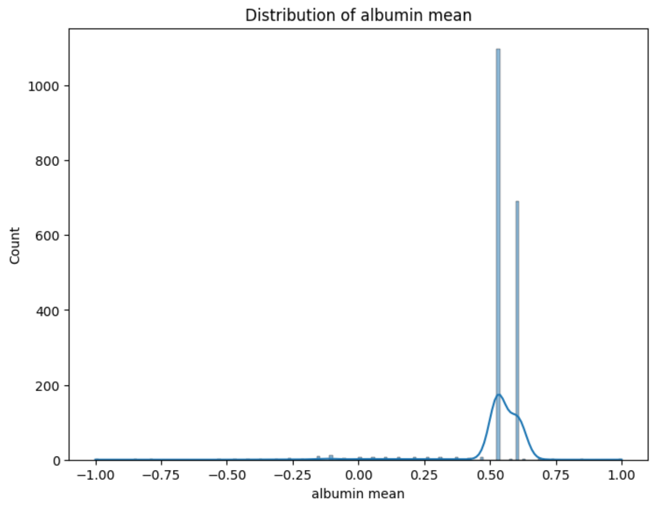
Now we go through all other columns which are supposed to be numerical, find the values in each of them which are of wrong type as string or other datatypes and replace them with mean value of the entire column.

Then we fill the missing values using median.

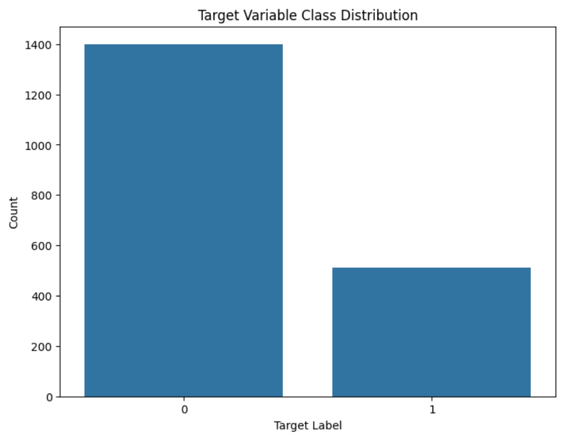
- Description of data and studying their distributions using plots:
  - Some of the distribution plots for example:
    -





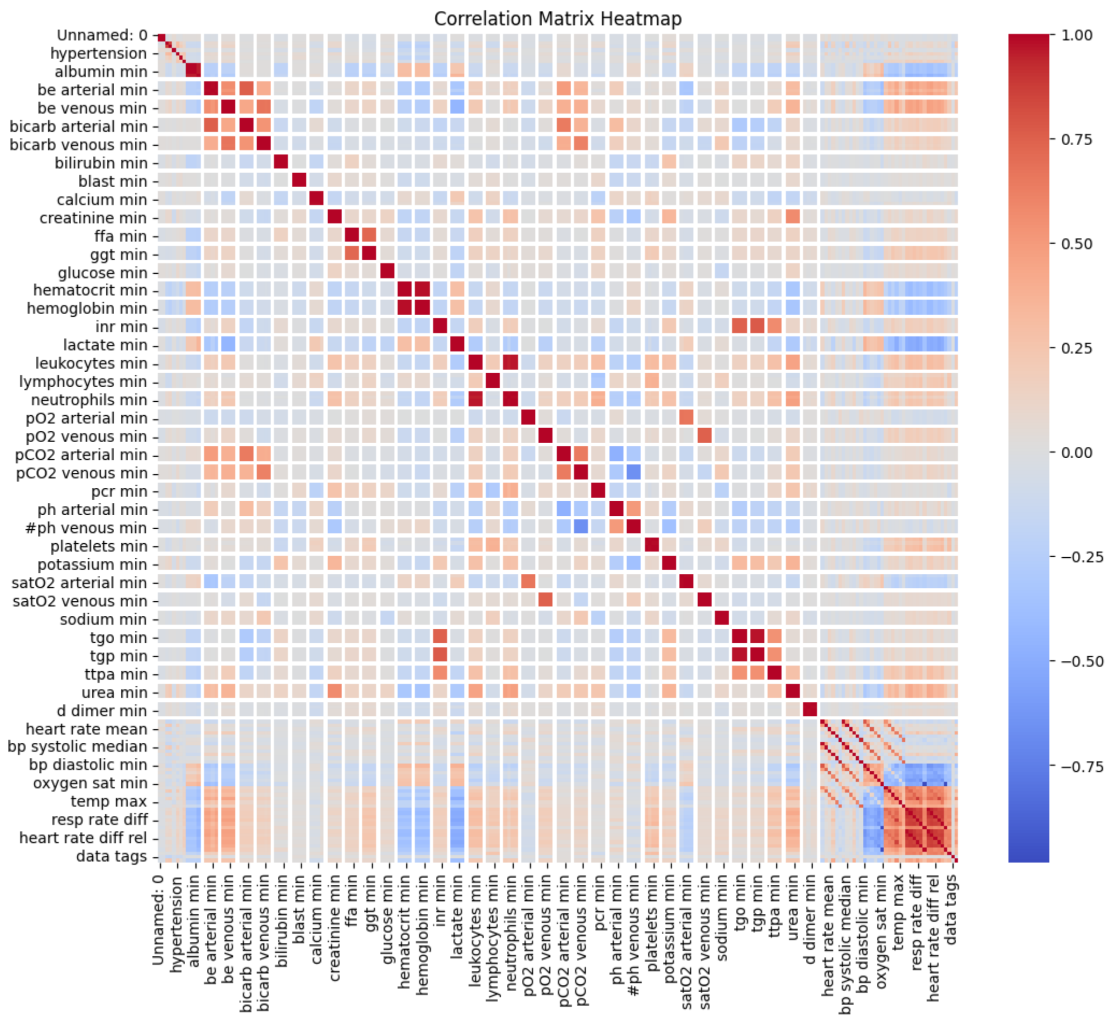


- Then we get the value of mean, median and mode for numerical columns.
- Then we check whether the dataset is balanced or imbalanced among different features.



-

The correlation matrix for further evaluation of features based on the strength of their relation with target value:



Then we proceed by deleting the features(columns) which has very low correlation with target variable (last column) to make the dataset more abstract:

- Set a threshold for correlation (e.g., 0.1)
- Get a list of features with correlation below the threshold
- Remove that features from the dataset

After splitting the data into test and train, because of unequal count of target classes, we should take samples in each turn of training in the way that the exposure of model with both classes be of equal count.

List of models to be tried:

- Logistic regression
- Decision tree classifier
- Random Forest classifier

- K neighbors’ classifier

Results of model selection:

The model with the best performance is: **Logistic Regression**:

Best Accuracy: 0.8512

Best Precision: 0.6810

Best Recall: 0.7980

Best F1-score: 0.7349

Best ROC AUC: 0.8932

Evaluation of performance of best model after final training:

Best F1-score: 0.7407

Best ROC AUC: 0.9051

Best Hyperparameters: {'C': 10, 'penalty': 'l1', 'solver': 'liblinear'}

Challenges I faced:

- Too much missing values!
- Too much mistaken values (values of wrong data type)!
- Irrelevant values in each column
- Lack of balance between two target classes
- Presence of string or object data type values in numerical columns