Cairo University

Faculty of Engineering

[CMPS451- Data Mining,Big Data and Data Analytics]

Patient survival prediction for cases admission

Team Members :

Yasmin Hashem Niazy 4200014

Sohad Hossam ELdin 1190019

Bassant Hisham 1190018

Yasmin Zaki Bassiouny 1190352

Table Of Contents:

1. Idea:

We aim to develop and validate a prediction model for all-cause in-hospital mortality among admitted patients. Utilising machine learning techniques, we will analyse a dataset containing various factors associated with hospitalizations to predict whether a patient will survive or not decide on their admission .

2. Business Value :

1. Enhanced Patient Care: Accurate prediction of patient survival upon admission allows for timely interventions, improving patient outcomes and satisfaction.

2. Resource Efficiency: Predictive models optimise resource allocation, including staffing and equipment, leading to cost savings and better resource utilisation.

3. Cost Reduction: Early identification of high-risk patients reduces healthcare costs by preventing adverse events and shortening hospital stays.

4. Data-Driven Decisions: Machine learning enables data-driven decision-making, tailoring treatment plans based on individual patient risk profiles.

5. Research and Innovation: The project fosters research and innovation in healthcare analytics, advancing predictive modelling for improved patient outcomes.

6. Competitive Advantage: Organisations using predictive analytics gain a competitive edge by offering high-quality, data-driven healthcare services.

3. Technical Part

3.1 Data Preprocessing :

3.1.1 Data Cleaning:

In the preprocessing stage of our project, we implemented several steps to ensure the quality and reliability of our data. Here's a detailed overview of the processes we performed:

1. Removing Missing Data: We started by identifying and removing instances with missing values, N/A entries, and unexpected data points. This step ensures that our dataset is clean and suitable for analysis. We also got rid of columns we know won't be useful for further processing in the project .
2. Dealing with Outliers: We addressed outliers in our dataset to prevent them from skewing our analysis. Specifically, we focused on columns related to ICU and hospital death probabilities, filtering out values below 0 to maintain data integrity.
3. Data Normalisation/Standardization: At this stage, we planned to perform data normalisation or standardisation to ensure that all features are on a similar scale. However, this step is currently pending implementation.
4. Transforming Features to Categorical: We transformed categorical features as needed to prepare them for further analysis. For binary string columns, we converted them to integer values (0s and 1s) for compatibility with machine learning algorithms.
5. Visualising the Data: We visualised the cleaned dataset using various plots, including box plots, scatter plots, and histograms. These visualisations provide insights into the distribution and relationships between different features, aiding in further analysis and model development.

3.1.2 Implementation:

The above preprocessing steps were implemented using PySpark, a powerful tool for big data processing. We divided our dataset into chunks to parallelize the preprocessing tasks, ensuring efficient processing even on large datasets. The `preprocess_map` function handled the preprocessing logic for each chunk, while the `preprocess_reduce` function combined the processed chunks into a single DataFrame.

Furthermore, our custom my_map and my_reduce functions facilitated the map-reduce approach, allowing us to apply the preprocessing steps in a distributed manner. Finally, the cleaned dataset was converted to a Pandas DataFrame for further analysis and modelling.

3.2 EDA and Data Visualization

3.2.1 Introduction to Data Visualization:

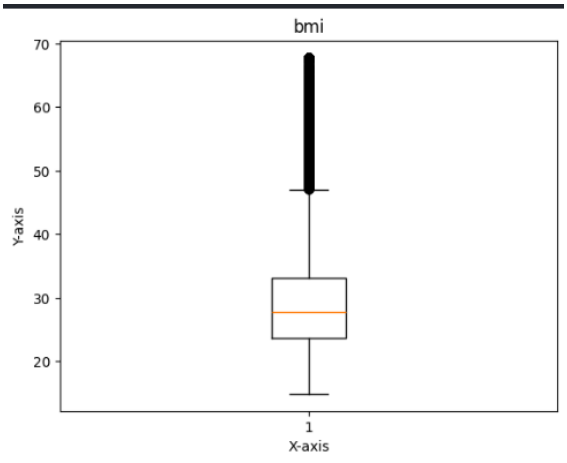
- Data visualisation plays a crucial role in uncovering insights and patterns within datasets. By transforming raw data into visual representations, we gain a deeper understanding of the underlying structure and relationships within the data.

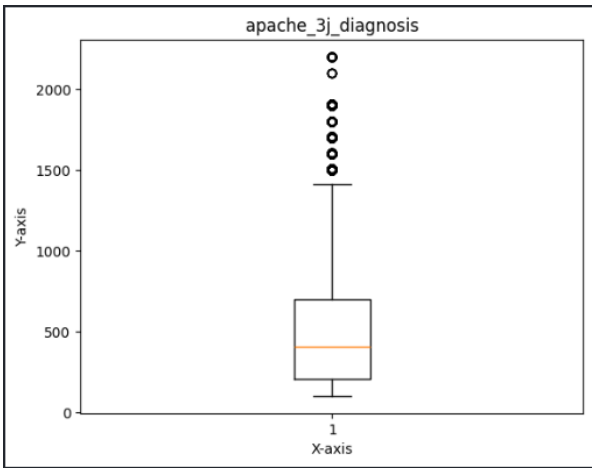
3.2.2 Boxplots and Histograms:

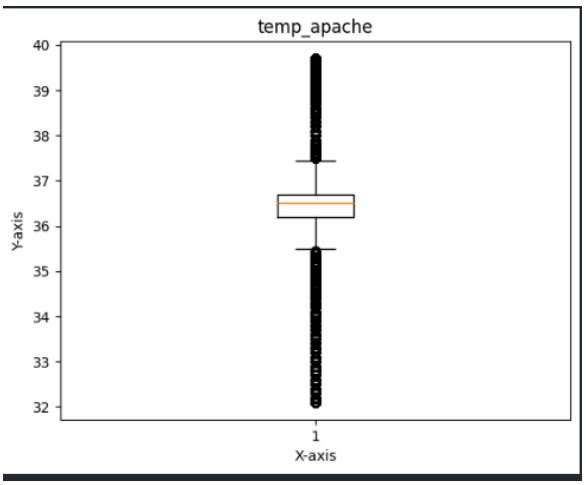
- Boxplots and histograms are powerful tools for visualising the distribution of numerical variables in our dataset.
- Histograms display the frequency distribution of data, allowing us to observe the shape and density of values.

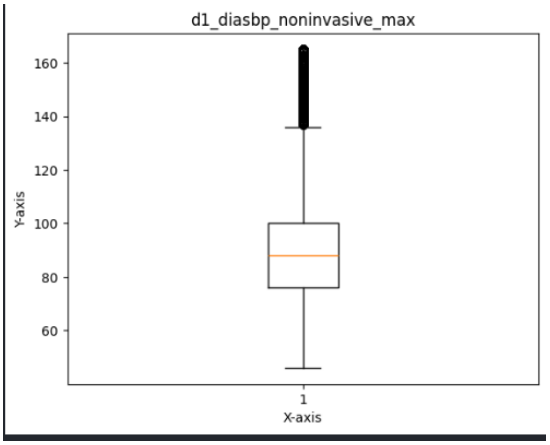
3.2.3 Box Plots:

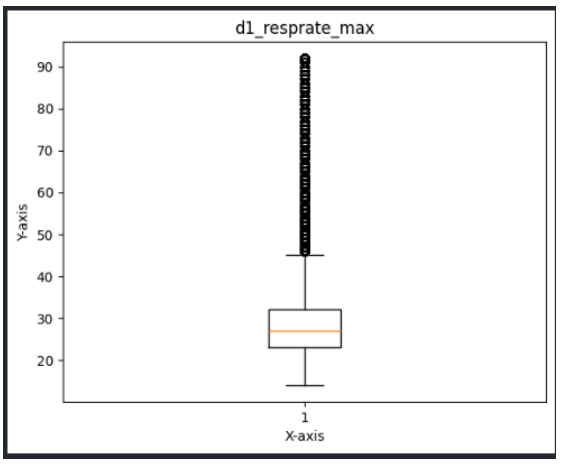
- We used boxplots to observe the outliers and to analyse the range of values of the features of the dataset so as to remove any values that are out of range in the features.

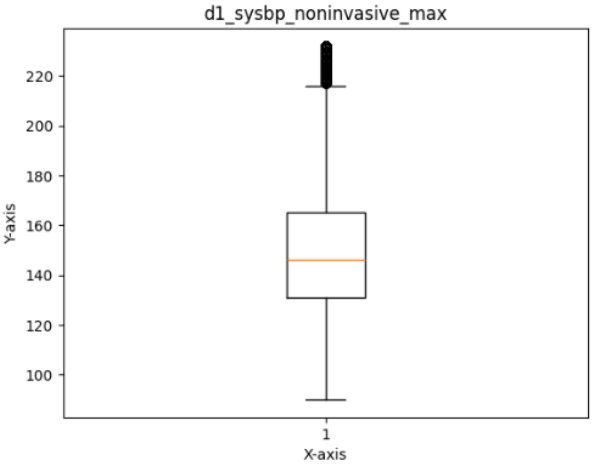






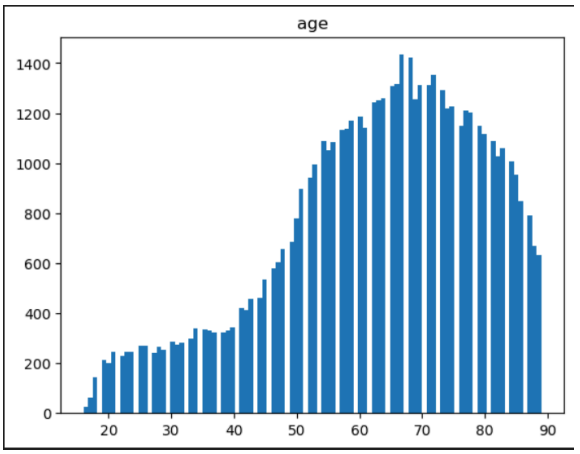


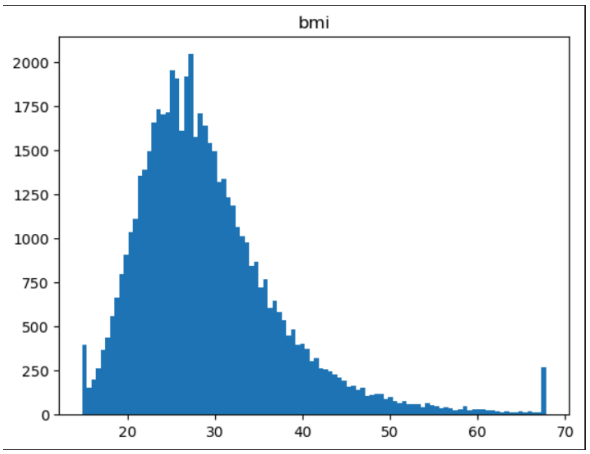


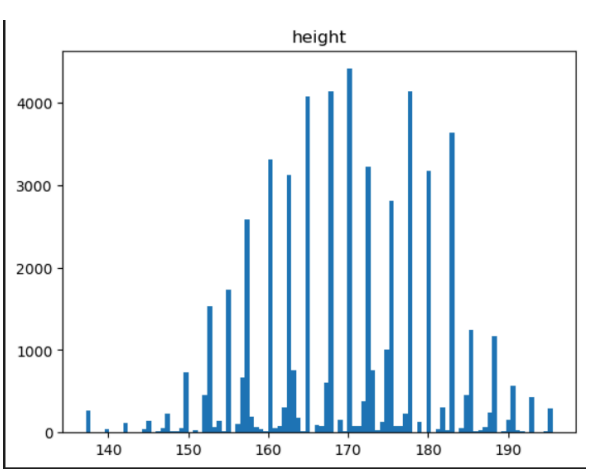


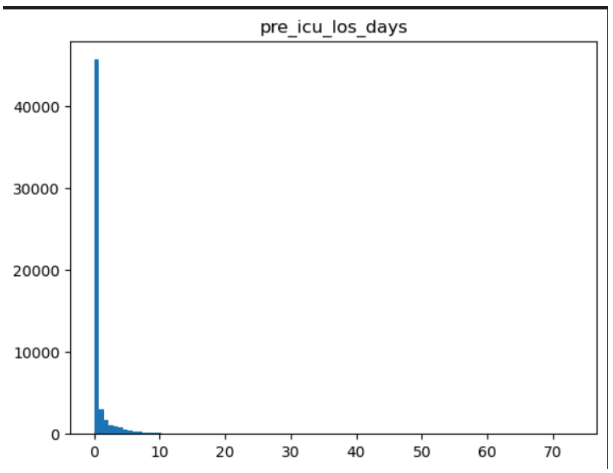
3.2.4 Histograms :

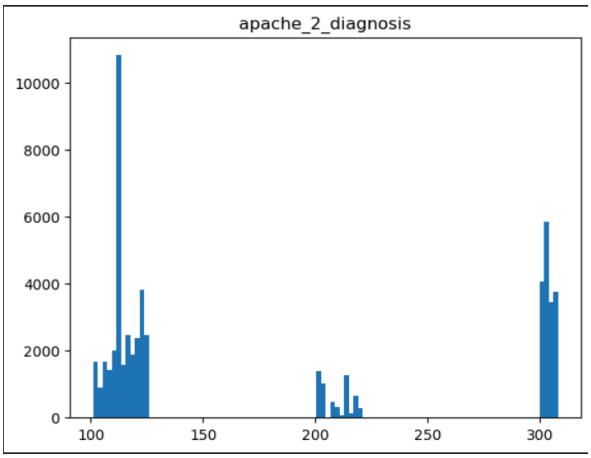
- We used histograms to analyze the range of values of different columns , and see their frequency and distribution

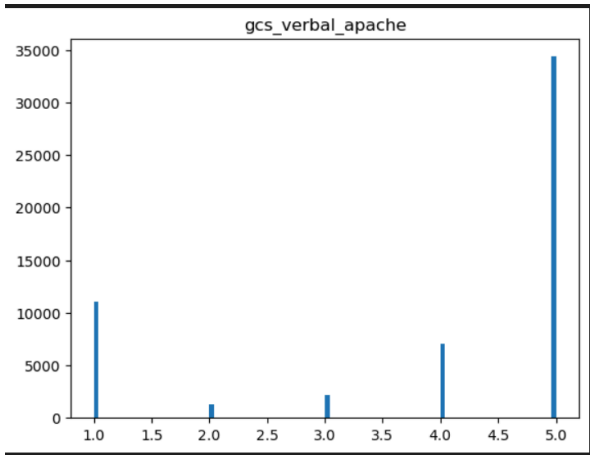






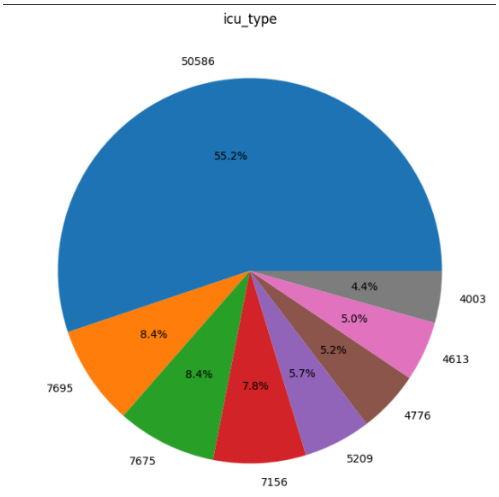


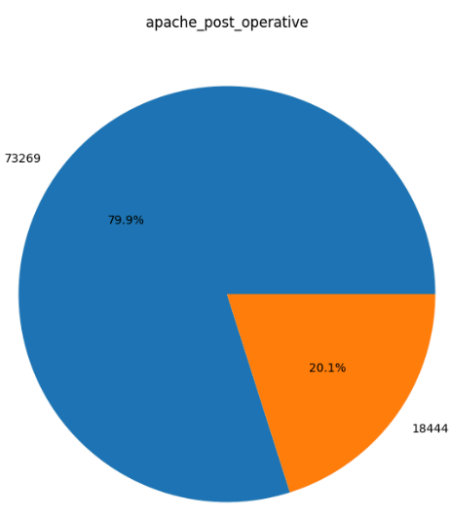


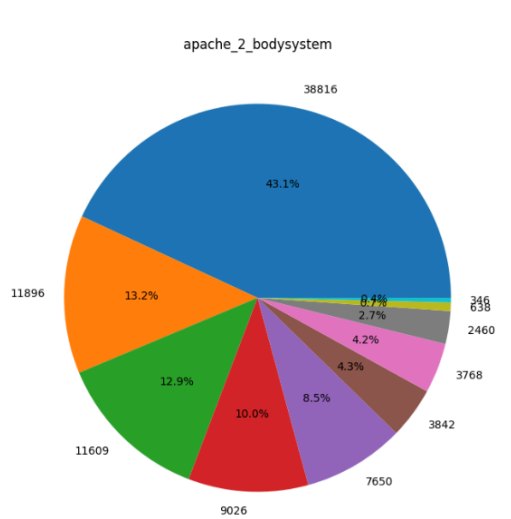


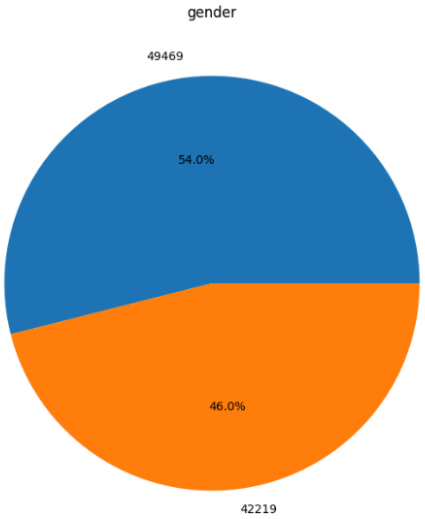
3.2.5 Pie Charts:

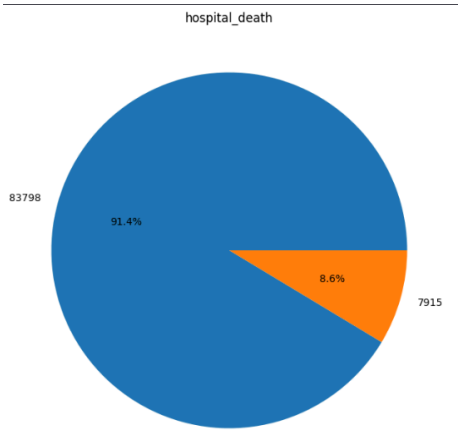
- Pie charts gave us a better perspective on the contents of the data set and the columns with many variations of data types

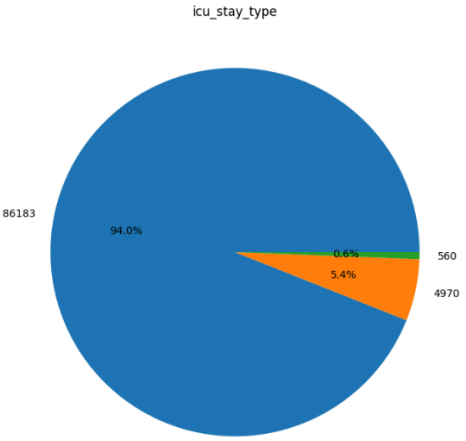






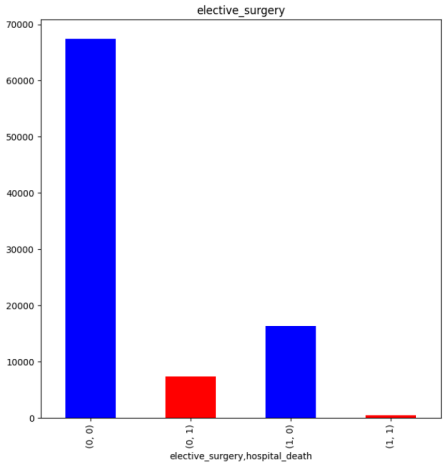


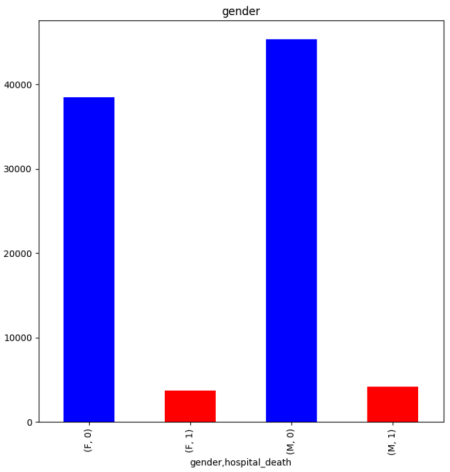


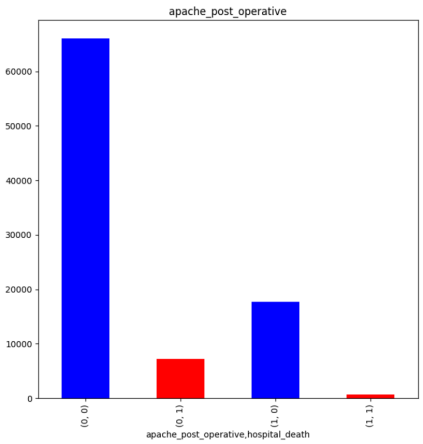


3.2.6 bar Charts:

- Bar charts gave us a better perspective on the relation between different features values and hospital deaths.







3.3 Model Building and Predictions

3.3.1. `my_map` Function:

- This function applies a given function (`map_func`) to each element in an iterable in parallel using threads.
- It takes two arguments: `func` (the function to apply) and `iterable` (the iterable object).
- Within this function, each element of the iterable is processed concurrently using threads, and the results are collected into a dictionary (`mapped_iterable`).

3.3.2. `map_func` Function:

- This function represents the mapping step of parallel processing.
- It takes a chunk of data and a classifier as input.

- It preprocesses the data, trains the classifier, and evaluates its performance using metrics such as accuracy, precision, and recall.
- The results (true labels and predicted labels) are returned in a dictionary format.

3.3.3. `my_reduce` Function:

- This function aggregates the results obtained from the mapping step.
- It combines the results from different threads into a single result using a reduction function (`reduce_func`).
- The aggregated result is returned as a dictionary.

3.3.4. `reduce_func` Function:

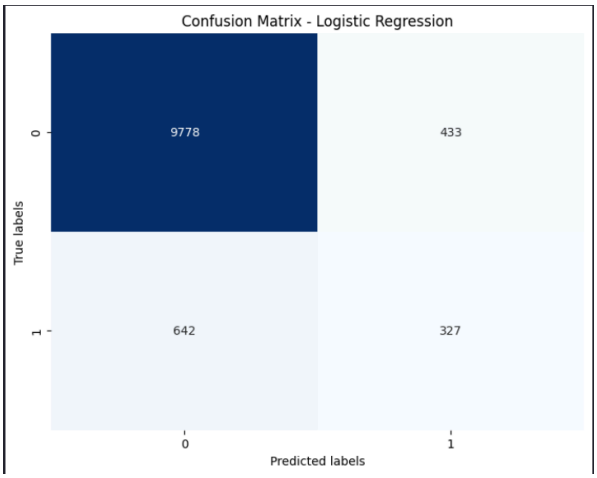
- This function represents the reduction step of parallel processing.
- It combines the results obtained from individual threads into a consolidated result.
- In this case, it merges the true labels and predicted labels obtained from different chunks of data.

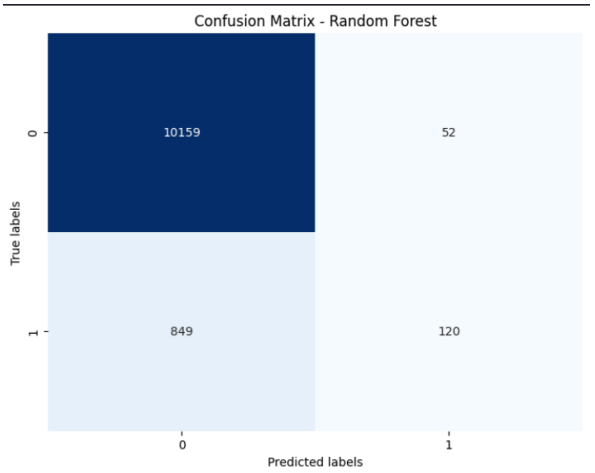
3.3.5. Classifiers and Dataset Partitioning:

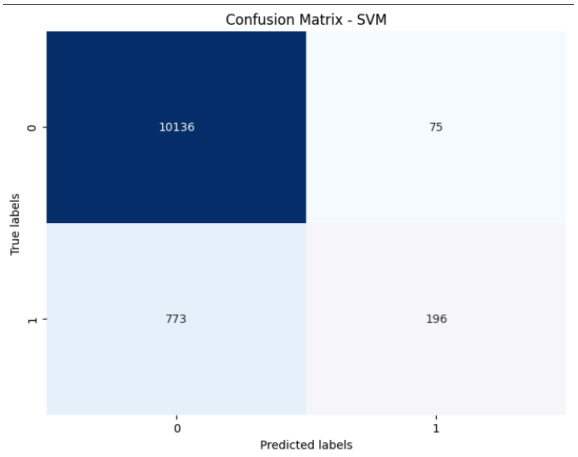
- Several classifiers (Logistic Regression, Random Forest, SVM,KNN,Gradient Boosting, Decision Trees) are defined along with their configurations.
- The dataset is partitioned into chunks to facilitate parallel processing.
- Each chunk of data is processed independently by the `map_func` function.

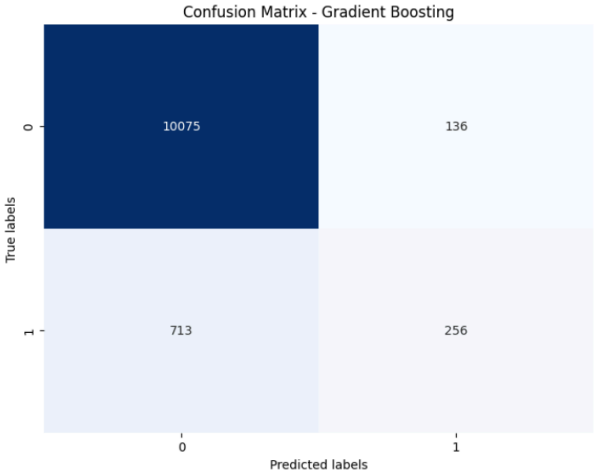
3.3.6. Evaluation and Visualization:

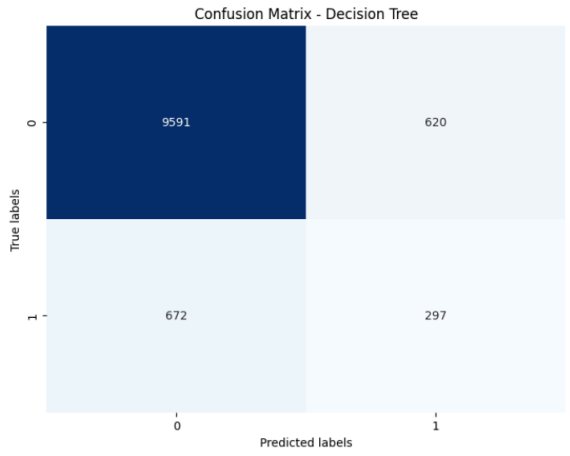
- Performance metrics such as accuracy, F1 score, and recall are calculated for each classifier using the true and predicted labels.
- Confusion matrices are plotted for each classifier to visualize their performance.
- Finally, the performance metrics are aggregated and visualized using bar charts, allowing for easy comparison between different classifiers.

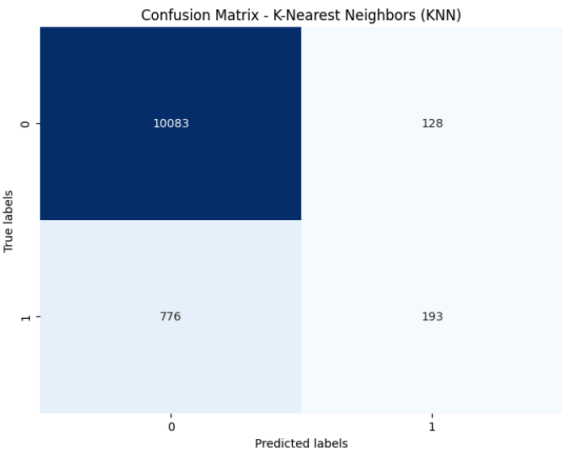

Confusion Matrix - Logistic Regression
9778
433
642
327
True labels
Predicted labels
0
1


Confusion Matrix - Random Forest
10159
52
849
120
True labels
Predicted labels
0
1


Confusion Matrix - SVM
10136
75
773
196
True labels
Predicted labels
0
1


Confusion Matrix - Gradient Boosting
10075
136
713
256
True labels
Predicted labels
0
1


Confusion Matrix - Decision Tree
9591
620
672
297
True labels
Predicted labels
0
1


Confusion Matrix - K-Nearest Neighbors (KNN)
10083
128
776
193
True labels
Predicted labels
0
1

Followed by visiualizing the Difference in the resulted accuracy , recall and F1 score from all the 6 ML models