KDDM - Final Report On Medecine Dataset

Monday 24th February, 2025 - 12:58

Koch Elliot
University of Luxembourg
Email: kochelliotpro@gmail.com

1. Introduction

Diabetes, a prevalent chronic metabolic disorder, poses significant health challenges worldwide. Accurate prediction of diabetes in patients can help in early intervention and improve patient outcomes.
This report highlights a project that harnesses the power of machine learning techniques to develop a predictive model for identifying patients with diabetes. To accomplish this, we used a medical dataset that included information from diabetic and non-diabetic patients.
The primary goal was to leverage advanced machine learning algorithms to create a model capable of determining patients who have diabetes and those who do not.

2. Data Visualization

The first phase involved visualizing the dataset to gain a better understanding. We examined the columns, which represented various types of data related to the patients. The dataset is about 50 columns, including patient information (18 columns), medical information (31 columns), and a column indicating the type of diabetes (diabetesMed).

To identify relevant information, remove unusable data, and identify outliers, we plotted different types of data in different ways.
First, we plotted the data based on ethnicity and gender. From the ethnicity plot, we observed that the dataset predominantly included Caucasian and African-American patients. While it provided information about race and location, it did not have significant medical implications. The gender plot showed that there were more female patients than male patients, suggesting a gender proportion imbalance (small). See Figure 1 and See Figure 2,

Next, we examined the age and weight distributions for male patients. We noticed a difference in the number of entries between the age and weight plots (12'000 / 700). We then plotted the age and weight distributions for female patients, which revealed a similar discrepancy in entries. See Figure 3 and See Figure 4

Further analysis involved plotting the number of entries per admission type. This revealed a substantial difference between the first three admission types (Emergency, Urgent, and Elective) and the others. We observed a significant number of NULL values for the admission type and a negligible amount for the newborn admission type. See Figure 5

We also plotted the discharge disposition, which exhibited a similar pattern. The entries corresponding to Discharged to Home, Discharged to SNF, and Discharged to Home with Home Health Services were predominant, while NULL values were more present than for the other categories. See Figure 6

Finally, we plotted the admission source, revealing a similar pattern with two categories (Emergency Room and Physician Referral) having a large number of entries. NULL was the third most prevalent value. See Figure 5

Overall, these visualization steps provided insights into the dataset, allowing for better understanding and informed decision-making for subsequent data processing and analysis.

3. Data Pre-processing

In the data pre-processing phase, we implemented several steps to prepare the datasets for training. We created two separate datasets for training purposes: dataset 1 and dataset 2.

For dataset 1, we removed columns that contained the most missing values, represented as "?" in the dataset. Additionally, we excluded columns that were deemed not significant for the analysis. The columns removed from dataset 1 were 'encounter_id', 'patient_nbr', 'weight', 'payer_code', and 'medical_specialty'. And finally, we selected patients from the dataset that was only from the most representative type for the Admission Type/Sources and Discharge Disposition columns. In dataset 2, we only removed the columns with the most missing values.
That was the only step that differs in the pre-processing for

both datasets.

Next, we replaced every entry marked as ”?” with NaN and dropped the rows containing NaN values. Removing the missing values helped clean the datasets from columns and rows with significant missing values.

The next step was to convert specific columns into a floating-point format using the "astype" method. And for the ’diabetesMed’ column, we replaced ”Yes” and ”No” with 1 and 0, respectively, using the "replace" method.

Additionally, we used One Hot Encoding to process columns with a string data type. This method created new columns for each label in the original column, assigning a value of 1 to rows corresponding to the respective class and 0 to others. For example, the ’insulin’ column became ’Insulin_Yes’ and ’Insulin_No’ columns.

Furthermore, we constructed a correlation matrix using the Spearman method that is more robust than the Pearson method. The correlation matrix allowed us to determine the best ”k” features/columns in relation to the target column, ’diabetesMed’. That enabled us to select a specific number of columns, such as 10 or 20, to include in the analysis. In this case, we choose 30 features.

After pre-processing, dataset 1 contained $63,685$ rows, while dataset 2 had $89,782$ rows out of the initial 101,766 data.
In summary, these pre-processing steps ensured that the datasets were formatted and prepared for subsequent analysis and training.

4. Training

In this section, we will explain what are the different techniques we applied to the data.

We implemented five different supervised learning techniques that are Linear Regression, Random Forests, Support Vector Machine (SVM), Decision Tree and Gradient Boosting Machine (GBM) and tried an unsupervised learning technique K-means. We focused more on Deep Learning with hyper-parameter optimization.

4.1. Supervised / Unsupervised Learning Results

For each of the supervised learning methods and K-means, we evaluate the results using the accuracy, the precision, the recall, the f1 score and a confusion matrix for both datasets. With those different values, we can compare them and determine which are the most effective.

Here are the different figures of those results: Figure 7. We see that the confusion matrices for each method are really similar and these are good results.

4.2. Deep Learning

We implemented a neural network. For this we used Keras. We first created a standard neural network with 4 layers. An input layer with 64 nodes, two hidden layers with 32 nodes and one output layer with 1 node. Each of them are dense layers, the first three layers use ReLu as activation function and the output layer uses a linear activation function. We then compile with Huber as the loss function and Adam as the optimizer. Finally, we trained it with 10 epochs and 64 patch size. We also have a Dropout of 0.5 on the third layer.

Here are the results for this basic model for the dataset 1:

- Test Loss: 0.0017
- Test Accuracy: 0.9960

Here are the results for this basic model for the dataset 2:

- Test Loss: 0.0028
- Test Accuracy: 0.9950

We then optimize the hyper-parameters which are the epochs, the batch size and the learning rate. For each parameters we tested multiple values. For the epochs, we tested 5, 10, 25 and 50, for the batch size, 32, 64, 128 and 256 and finally for the learning rate, 0.0001, 0.001, 0.01, 0.1. For each value, we output the loss and the accuracy and plot it. We then choose the best indeed, we want to have the lowest loss value and the highest accuracy.

Here are the hyper-parameters choice based on the losses and accuracy’s graphs for dataset 1:

- Epochs: 50, Figure 8
- Batch-Size: 32, Figure 9
- Learning Rate: 0.001, Figure 10

Here are the hyper-parameters choice based on the losses and accuracy’s graphs for dataset 2:

- Epochs: 50, Figure 11
- Batch-Size: 256, Figure 12
- Learning Rate: 0.001, Figure 13

And here are the results for this optimize model for dataset 1:

- Test Loss: 0.00157
- Test Accuracy: 0.9987

And here are the results for this optimize model for dataset 2:

- Test Loss: 0.0039
- Test Accuracy: 0.9959

5. Conclusion

In conclusion, this project focused on predicting diabetes in patients using a medical dataset.

The first step involved understanding the dataset and visualising important columns to gain insights into the variables. The next step was data pre-processing, which included handling missing values, applying One-Hot Encoding (O-H-E), and examining the correlation matrix. Additionally, all the data was normalised to ensure consistency.

During the prediction phase, a diverse set of machine learning methods were used, with a focus on Deep Learning techniques. To improve the performance of the Deep-Learning models used, we applied some hyper-parameter optimisation techniques.

The prediction results have the potential to aid in the early detection of diabetes, facilitate personalised healthcare strategies, and ultimately improve patient outcomes. Further research and refinement of these predictive models can lead to significant advancements in diabetes management and prevention.

6. Appendix

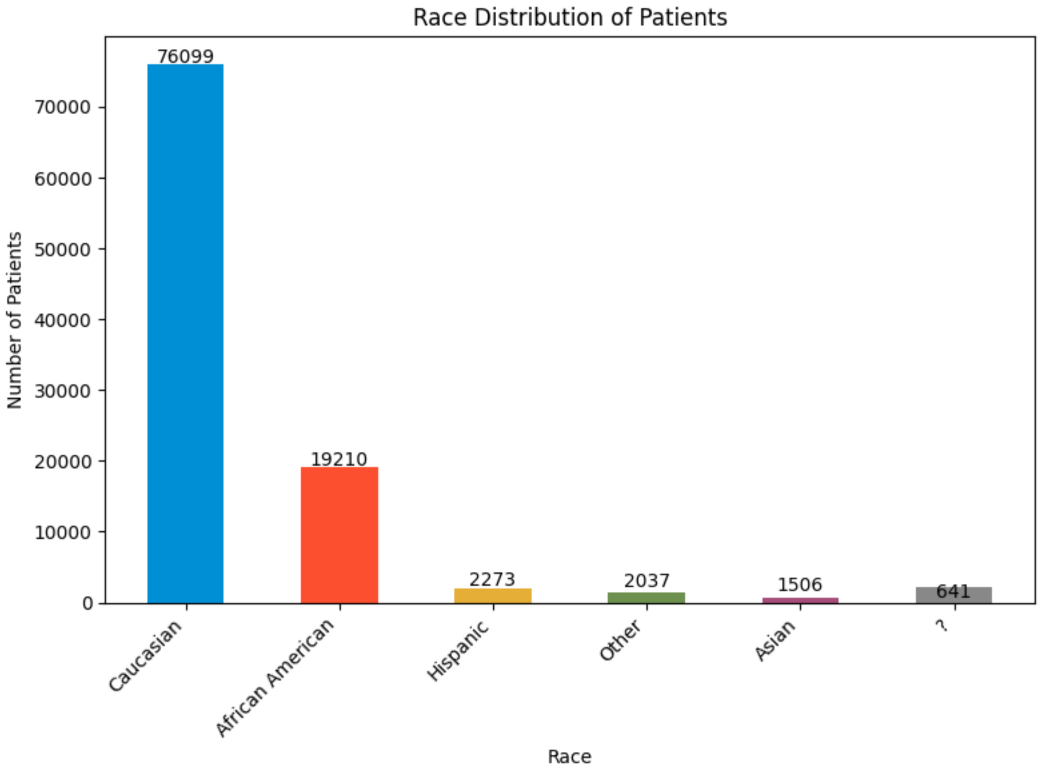


Figure 1. Ethnicity of the patients of the dataset

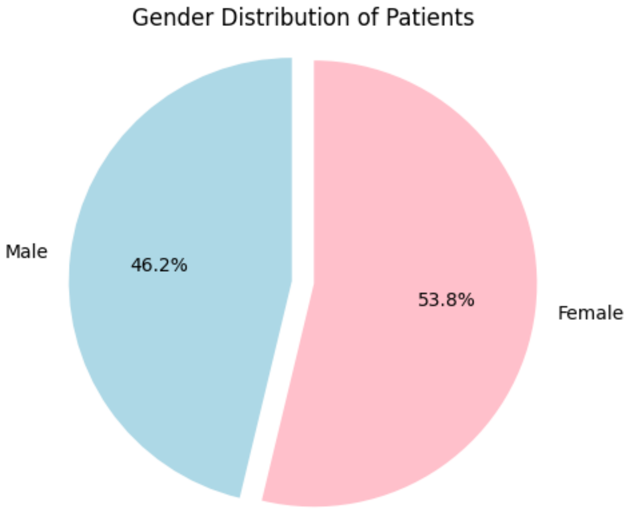


Figure 2. Gender distribution of the patients of the dataset

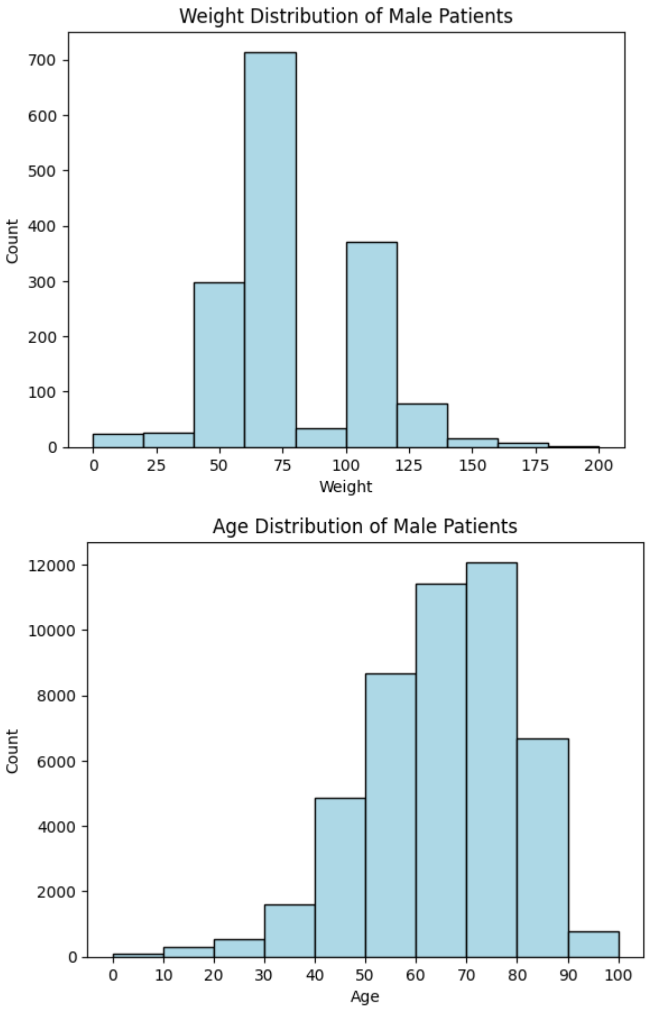


Figure 3. Age and Weight of the male patients of the dataset

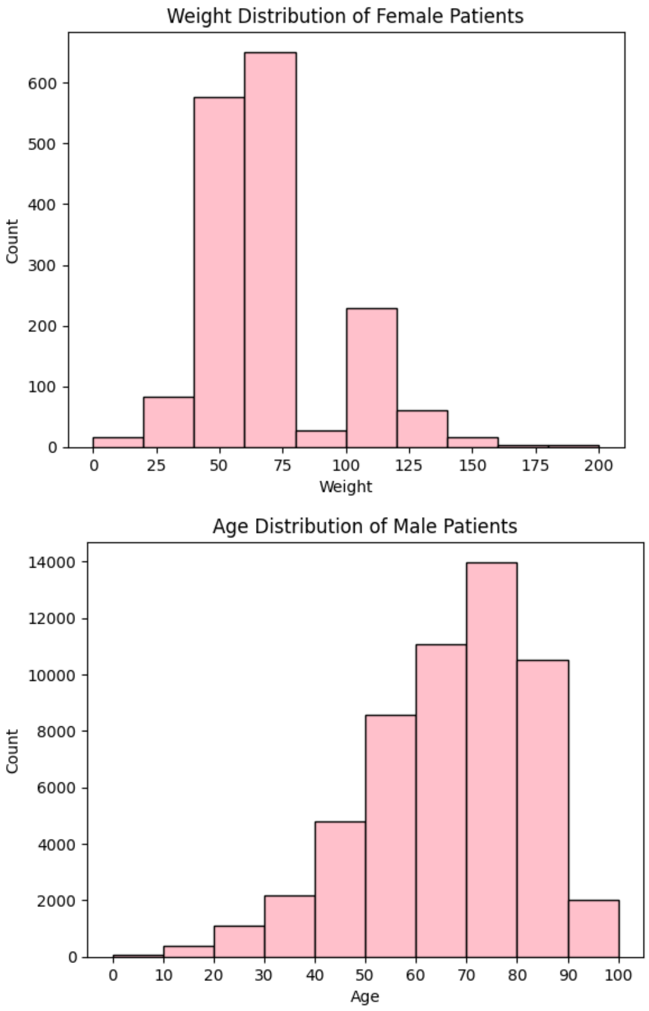


Figure 4. Age and Weight of the female patients of the dataset

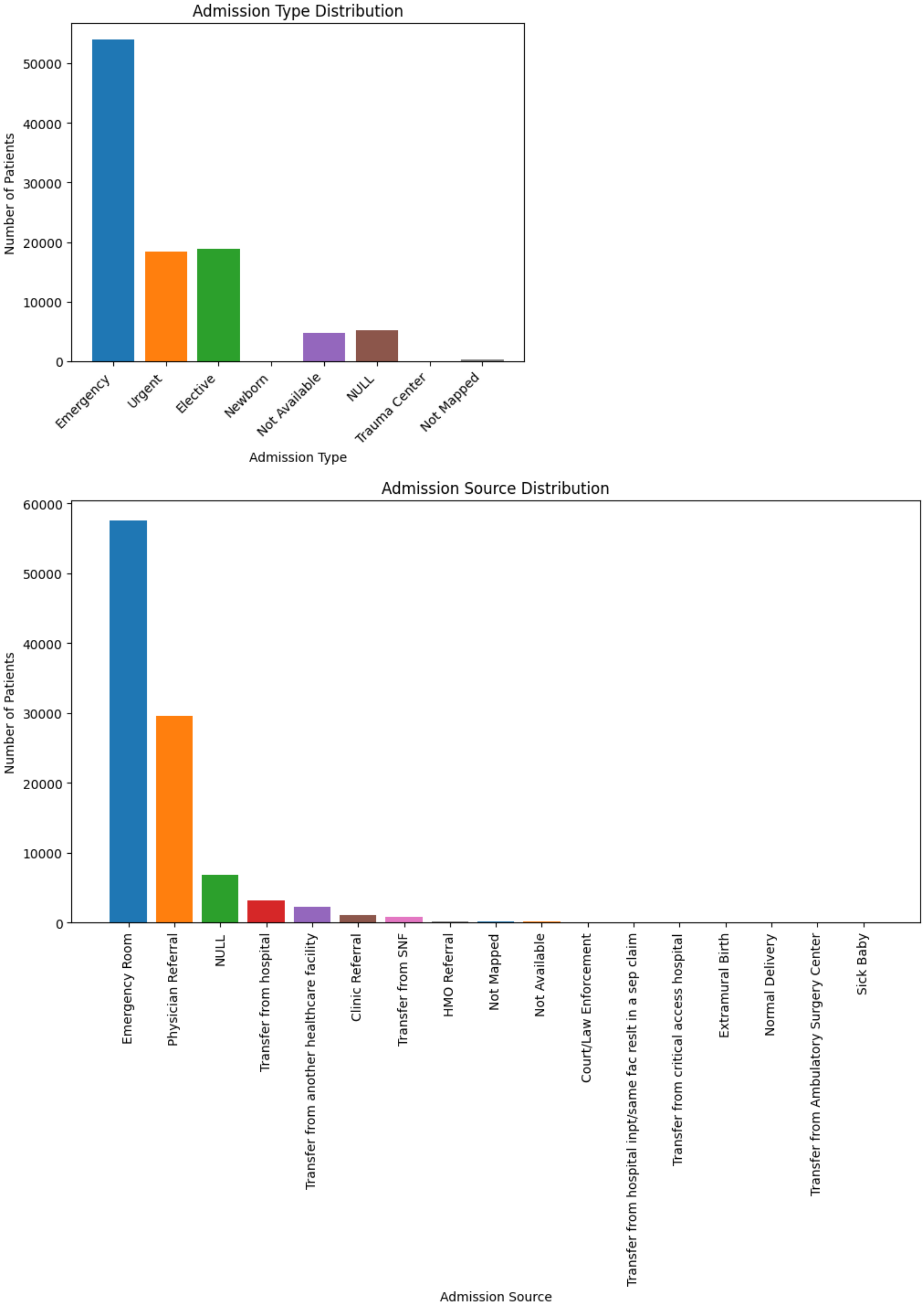


Figure 5. Admission Type/Source of the patients of the dataset

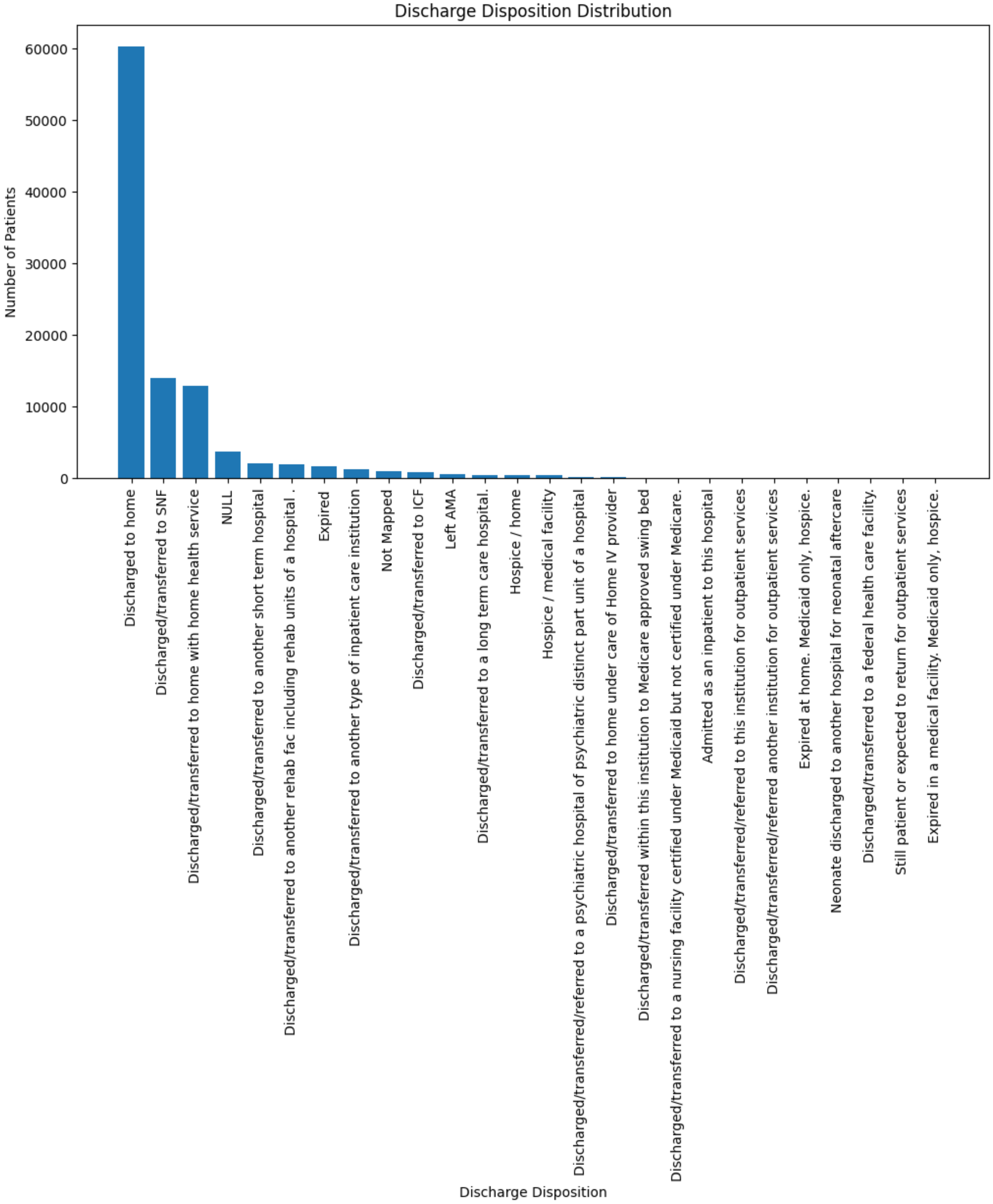
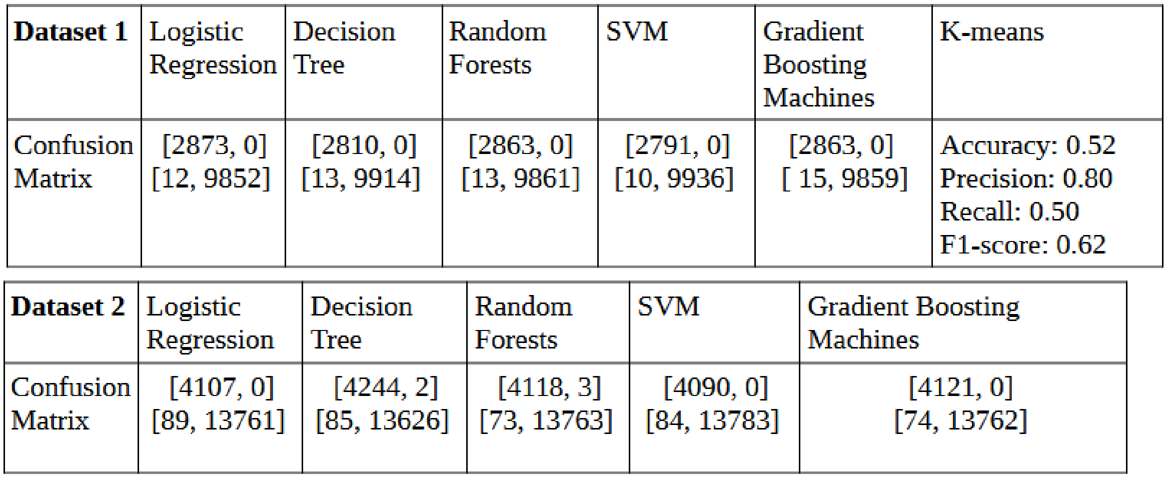


Figure 6. Admission Type/Source of the patients of the dataset



Dataset 1	Logistic Regression	Decision Tree	Random Forests	SVM	Gradient Boosting Machines	K-means
Confusion Matrix	[2873, 0] [12, 9852]	[2810, 0] [13, 9914]	[2863, 0] [13, 9861]	[2791, 0] [10, 9936]	[2863, 0] [15, 9859]	Accuracy: 0.52 Precision: 0.80 Recall: 0.50 F1-score: 0.62

Dataset 2	Logistic Regression	Decision Tree	Random Forests	SVM	Gradient Boosting Machines
Confusion Matrix	[4107, 0] [89, 13761]	[4244, 2] [85, 13626]	[4118, 3] [73, 13763]	[4090, 0] [84, 13783]	[4121, 0] [74, 13762]

Figure 7. Confusion matrix of the ML methods used for the training part of the project

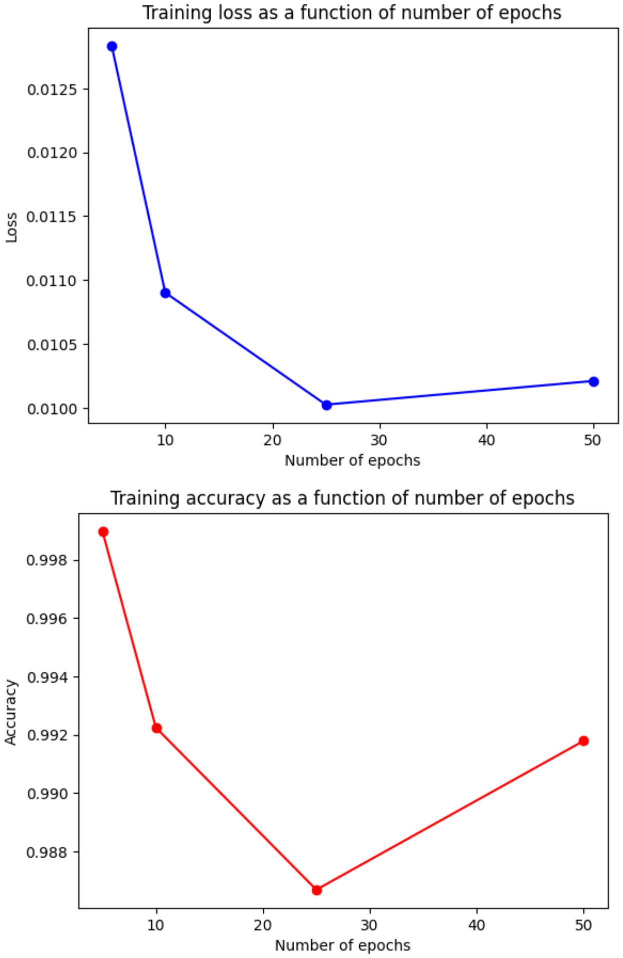


Figure 8. Loss and Accuracy of the model based on the Epochs

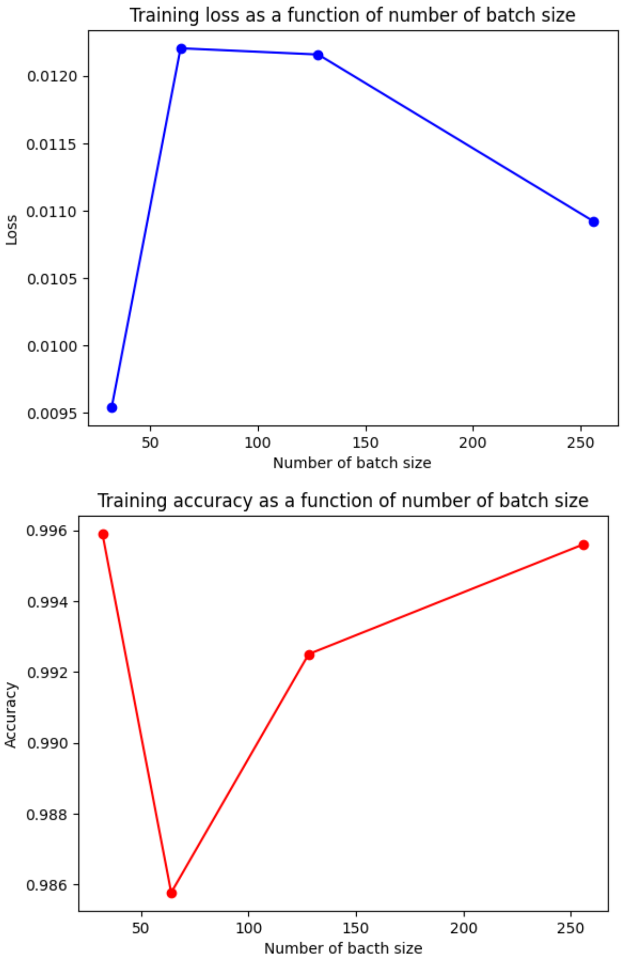


Figure 9. Loss and Accuracy of the model based on the Batch-Size

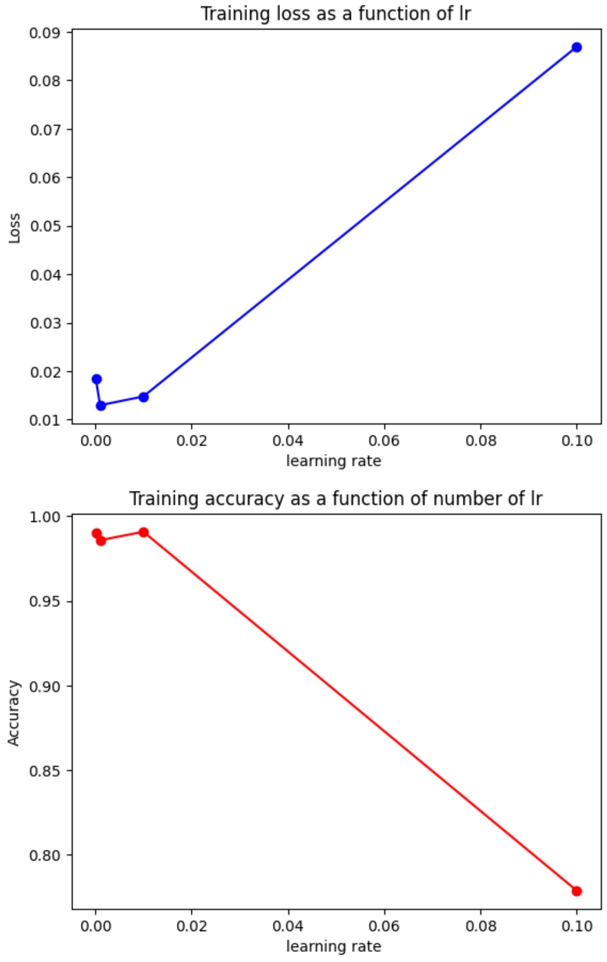


Figure 10. Loss and Accuracy of the model based on the Learning Rate

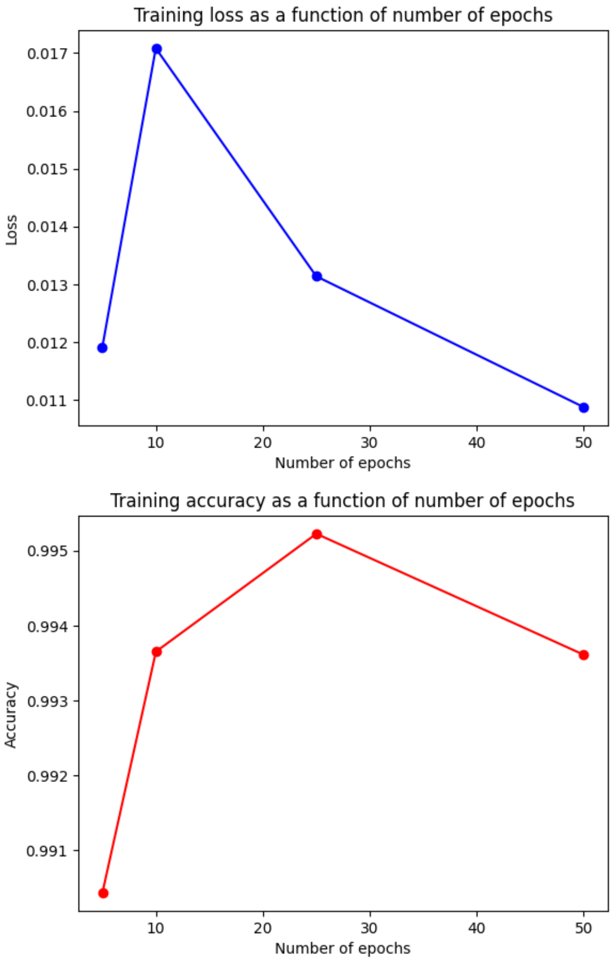


Figure 11. Loss and Accuracy of the model based on the Epochs

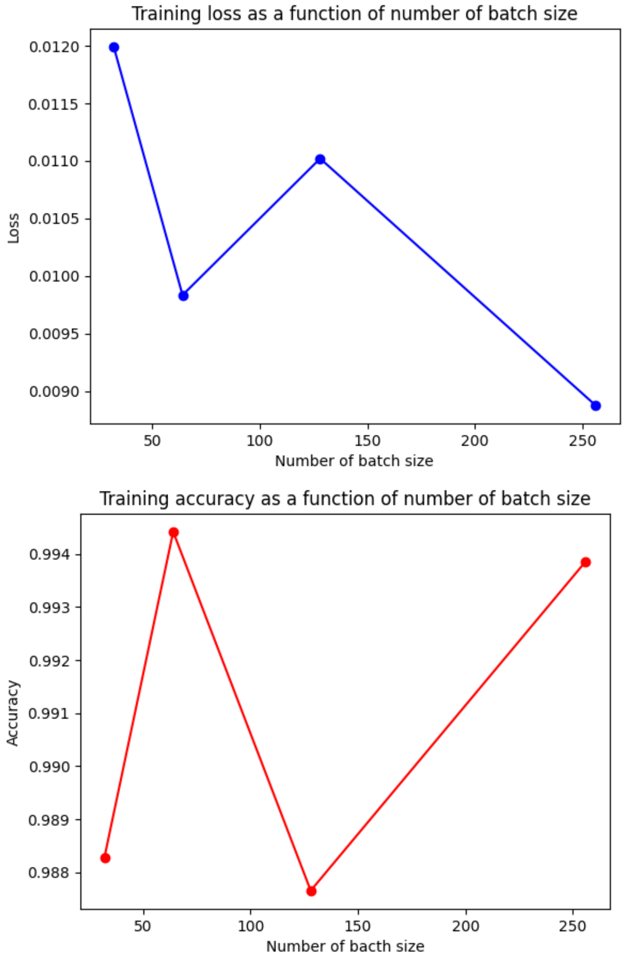


Figure 12. Loss and Accuracy of the model based on the Batch-Size

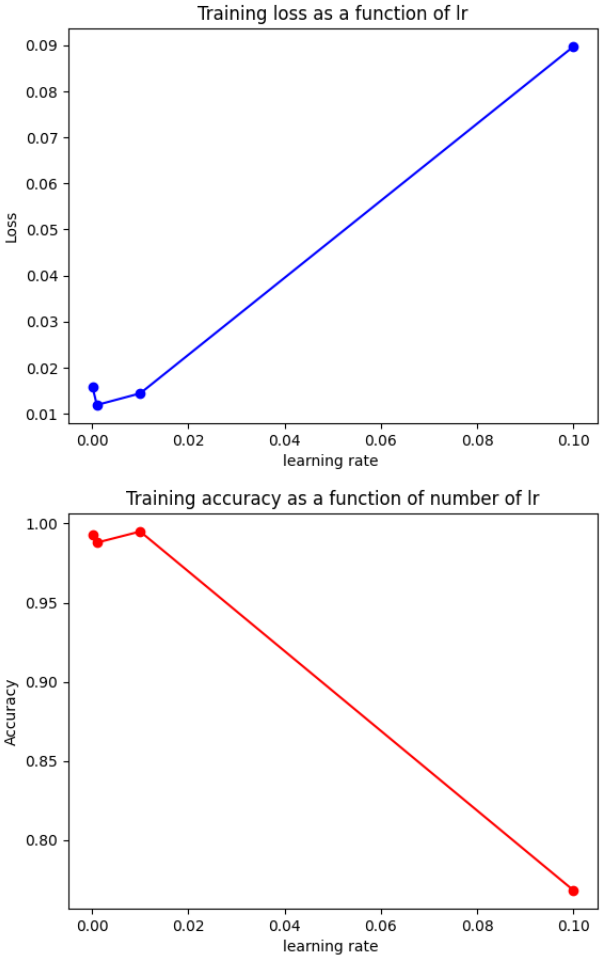


Figure 13. Loss and Accuracy of the model based on the Learning Rate