Reyhaneh Saffar _99222064

1. Loading Data

I used pandas to read the CSV file and store it in a data frame. The dataset has 110527 rows and 14 columns, containing information related to medical appointments and the factors influencing whether patients show up for their scheduled appointments.

1. Here is the description of each column:
   PatientId: A unique identifier for each patient (data type: numeric).
2. **AppointmentID**: A unique identifier for each appointment (data type: numeric).
3. **Gender**: The gender of the patient (data type: categorical: “M” for male, “F” for female).
4. **ScheduledDay**: The date when the appointment was scheduled (data type: datetime).
5. **AppointmentDay**: The date of the actual appointment (data type: datetime).
6. **Age**: The age of the patient (data type: numeric).
7. **Neighborhood**: The neighborhood where the medical facility is located (data type: categorical).
8. **Scholarship**: Indicates whether the patient receives financial aid (data type: binary: 0 or 1).
9. **Hypertension**: Indicates whether the patient has hypertension (data type: binary: 0 or 1).
10. **Diabetes**: Indicates whether the patient has diabetes (data type: binary: 0 or 1).
11. **Alcoholism**: Indicates whether the patient has alcohol-related issues (data type: binary: 0 or 1).
12. **Handicap**: Indicates the level of handicap (data type: numeric: 0, 1, 2, 3, or 4).
13. **SMSReceived**: Indicates whether the patient received an SMS reminder (data type: binary: 0 or 1).
14. **NoShow**: Indicates whether the patient showed up for the appointment (data type: binary: 0 or 1).

2. Data Preprocessing

- *Exploring Data*

  The Data Preprocessing section involved an initial exploration of the dataset to understand its structure and characteristics. By iterating through each column, unique values were identified and printed, providing insights into the range and distribution of data within each feature.

```
PatientId       [2.98724998e+13 5.58997777e+14 4.26296230e+12 ... 7.26331493e+13
 9.96997666e+14 1.55766317e+13]
AppointmentID   [5642903 5642503 5642549 ... 5630692 5630323 5629448]
Gender          ['F' 'M']
ScheduledDay    ['2016-04-29T18:38:08Z' '2016-04-29T16:08:27Z' '2016-04-29T16:19:04Z' ...
 '2016-04-27T16:03:52Z' '2016-04-27T15:09:23Z' '2016-04-27T13:30:56Z']
AppointmentDay  ['2016-04-29T00:00:00Z' '2016-05-03T00:00:00Z' '2016-05-10T00:00:00Z'
 '2016-05-17T00:00:00Z' '2016-05-24T00:00:00Z' '2016-05-31T00:00:00Z'
 '2016-05-02T00:00:00Z' '2016-05-30T00:00:00Z' '2016-05-16T00:00:00Z'
 '2016-05-04T00:00:00Z' '2016-05-19T00:00:00Z' '2016-05-12T00:00:00Z'
 '2016-05-06T00:00:00Z' '2016-05-20T00:00:00Z' '2016-05-05T00:00:00Z'
 '2016-05-13T00:00:00Z' '2016-05-09T00:00:00Z' '2016-05-25T00:00:00Z'
 '2016-05-11T00:00:00Z' '2016-05-18T00:00:00Z' '2016-05-14T00:00:00Z'
 '2016-06-02T00:00:00Z' '2016-06-03T00:00:00Z' '2016-06-06T00:00:00Z'
 '2016-06-07T00:00:00Z' '2016-06-01T00:00:00Z' '2016-06-08T00:00:00Z']
Age             [ 62  56   8  76  23  39  21  19  30  29  22  28  54  15  50  40  46   4
  13  65  45  51  32  12  61  38  79  18  63  64  85  59  55  71  49  78
  31  58  27   6   2  11   7   0   3   1  69  68  60  67  36  10  35  20
  26  34  33  16  42   5  47  17  41  44  37  24  66  77  81  70  53  75
  73  52  74  43  89  57  14   9  48  83  72  25  80  87  88  84  82  90
  94  86  91  98  92  96  93  95  97 102 115 100  99  -1]
Neighbourhood   ['JARDIM DA PENHA' 'MATA DA PRAIA' 'PONTAL DE CAMBURI' 'REPÚBLICA'
 'GOIABEIRAS' 'ANDORINHAS' 'CONQUISTA' 'NOVA PALESTINA' 'DA PENHA'
 'TABUAZEIRO' 'BENTO FERREIRA' 'SÃO PEDRO' 'SANTA MARTHA' 'SÃO CRISTÓVÃO'
 'MARUÍPE' 'GRANDE VITÓRIA' 'SÃO BENEDITO' 'ILHA DAS CAIEIRAS'
 'SANTO ANDRÉ' 'SOLON BORGES' 'BONFIM' 'JARDIM CAMBURI' 'MARIA ORTIZ'
 'JABOUR' 'ANTÔNIO HONÓRIO' 'RESISTÊNCIA' 'ILHA DE SANTA MARIA'
 'JUCUTUQUARA' 'MONTE BELO' 'MÁRIO CYPRESTE' 'SANTO ANTÔNIO' 'BELA VISTA'
 'PRAIA DO SUÁ' 'SANTA HELENA' 'ITARARÉ' 'INHANGUETÁ' 'UNIVERSITÁRIO'
 'SÃO JOSÉ' 'REDENÇÃO' 'SANTA CLARA' 'CENTRO' 'PARQUE MOSCOSO'
 'DO MOSCOSO' 'SANTOS DUMONT' 'CARATOÍRA' 'ARIOVALDO FAVALESSA'
 'ILHA DO FRADE' 'GURIGICA' 'JOANA D´ARC' 'CONSOLAÇÃO' 'PRAIA DO CANTO'
 'BOA VISTA' 'MORADA DE CAMBURI' 'SANTA LUÍZA' 'SANTA LÚCIA'
 'BARRO VERMELHO' 'ESTRELINHA' 'FORTE SÃO JOÃO' 'FONTE GRANDE'
 'ENSEADA DO SUÁ' 'SANTOS REIS' 'PIEDADE' 'JESUS DE NAZARETH'
 'SANTA TEREZA' 'CRUZAMENTO' 'ILHA DO PRÍNCIPE' 'ROMÃO' 'COMDUSA'
 'SANTA CECÍLIA' 'VILA RUBIM' 'DE LOURDES' 'DO QUADRO' 'DO CABRAL' 'HORTO'
 'SEGURANÇA DO LAR' 'ILHA DO BOI' 'FRADINHOS' 'NAZARETH' 'AEROPORTO'
 'ILHAS OCEÂNICAS DE TRINDADE' 'PARQUE INDUSTRIAL']
Scholarship     [0 1]
Hipertension    [1 0]
Diabetes        [0 1]
Alcoholism      [0 1]
Handcap         [0 1 2 3 4]
SMS_received    [0 1]
No-show         ['No' 'Yes']
```

Subsequently, for columns with categorical data types, detailed value counts were generated to understand the frequency distribution of each category. This allowed for a deeper understanding of the categorical variables, including gender, scheduled day, appointment day, neighborhood, and attendance status. These value counts provided crucial insights into the distribution and imbalance within the categorical variables. The picture below shows the part of output of this section.

```
Gender
F    71840
M    38687
Name: count, dtype: int64
ScheduledDay
2016-05-06T07:09:54Z    24
2016-05-06T07:09:53Z    23
2016-04-25T17:18:27Z    22
2016-04-25T17:17:46Z    22
2016-04-25T17:17:23Z    19
                        ..
2016-05-02T09:53:25Z     1
2016-05-30T09:12:28Z     1
2016-05-16T09:10:04Z     1
2016-05-09T10:17:48Z     1
2016-04-27T13:30:56Z     1
Name: count, Length: 103549, dtype: int
AppointmentDay
2016-06-06T00:00:00Z    4692
2016-05-16T00:00:00Z    4613
2016-05-09T00:00:00Z    4520
2016-05-30T00:00:00Z    4514
2016-06-08T00:00:00Z    4479
2016-05-11T00:00:00Z    4474
2016-06-01T00:00:00Z    4464
2016-06-07T00:00:00Z    4416
2016-05-12T00:00:00Z    4394
2016-05-02T00:00:00Z    4376
2016-05-18T00:00:00Z    4373
2016-05-17T00:00:00Z    4372
2016-06-02T00:00:00Z    4310
2016-05-10T00:00:00Z    4308
2016-05-31T00:00:00Z    4279
2016-05-05T00:00:00Z    4273
2016-05-19T00:00:00Z    4270
2016-05-03T00:00:00Z    4256
2016-05-04T00:00:00Z    4168
2016-06-03T00:00:00Z    4090
2016-05-24T00:00:00Z    4009
2016-05-13T00:00:00Z    3987
2016-05-25T00:00:00Z    3909
2016-05-06T00:00:00Z    3879
2016-05-20T00:00:00Z    3828
2016-04-29T00:00:00Z    3235
2016-05-14T00:00:00Z      39
Name: count, dtype: int64
Neighbourhood
JARDIM CAMBURI                 7717
MARIA ORTIZ                    5805
RESISTÊNCIA                    4431
JARDIM DA PENHA                3877
ITARARÉ                        3514
                               ...
ILHA DO BOI                      35
ILHA DO FRADE                    10
AEROPORTO                         8
ILHAS OCEÂNICAS DE TRINDADE       2
PARQUE INDUSTRIAL                 1
```

- *Handling Missing Values*
  Initially, the code identifies and counts the missing values for each column.

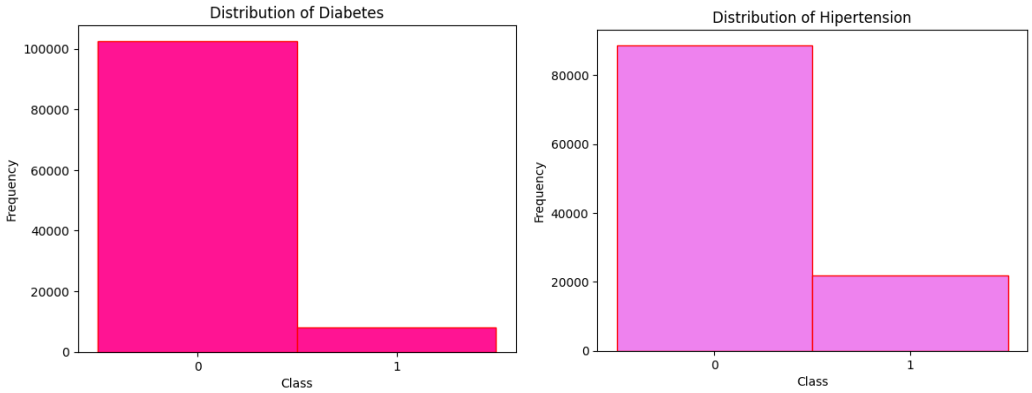
```
Missing values:
PatientId         0
AppointmentID     0
Gender            0
ScheduledDay      0
AppointmentDay    0
Age               0
Neighbourhood     0
Scholarship       0
Hipertension      0
Diabetes          0
Alcoholism        0
Handcap           0
SMS_received      0
No-show           0
dtype: int64
```

  As we can see, the dataset has no missing or null Value, but to ensure, i fill in missing values using appropriate strategies: for categorical variables, the most frequent value (mode) is imputed, while for numerical variables, the mean value is used. This ensures that the dataset is cleaned and ready for further analysis, with all missing values appropriately handled.

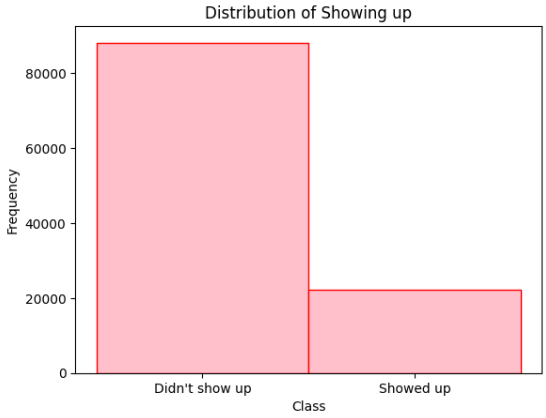
  In following, the 'AppointmentDay' and 'ScheduledDay' columns contained datetime information. Initially, these columns were converted into a standardized datetime format to facilitate easier manipulation and analysis of temporal data. Then, the datetime values were split into separate columns for date ('AppointmentDate' and 'ScheduledDate') and time ('AppointmentTime' and 'ScheduledTime'). This separation allowed for more granular analysis and comparison of appointment and scheduling times. The 'Age' column was examined for any anomalies or missing values. Entries with unrealistic ages (such as -1 or 0) were identified as potential errors. To handle these anomalies, I replaced entries with ages of -1 or 0 with the mean age calculated from the dataset. This ensured that unrealistic age values did not skew the analysis. After replacing the invalid ages, the dataset was checked to confirm that no entries with -1 or 0 remained in the 'Age' column. Columns such as 'PatientId' and 'AppointmentID' were deemed unnecessary for the subsequent statistical analysis. These columns were removed from the dataset to streamline the analysis and reduce unnecessary data overhead.

- Visualization Of Imbalance Classes

In this section, visualization of class imbalances was conducted to gain insights into the distribution of different classes within categorical variables. The code iterated through each column in the dataset, identifying binary categorical variables (values of 0 and 1). For each binary variable found, a histogram was plotted to display the frequency of each class (0 and 1). This visualization enabled a clear understanding of the distribution of classes within each binary variable. Here is some of the plots:



Additionally, a separate histogram was created specifically for the target variable 'No-show', which indicates whether a patient showed up for their appointment or not.



3. New Feature

In the New Feature section, a new feature called 'TimeDifference' was engineered from existing date-related columns in the dataset. Specifically, the difference in days between the appointment date and the scheduled date was calculated and assigned to the new feature column. This process leveraged the datetime functionalities available in pandas to compute the time difference efficiently.

The newly created 'TimeDifference' feature provides valuable information about the gap between scheduling an appointment and the actual appointment date. However, we have values like -1 or -6 in 'TimeDifference' column which we handled them in furture steps.

4. Encoding

In the Encoding section, the categorical variables within the dataset were transformed into numerical representations. The 'No-show' variable, indicating whether a patient attended their appointment, was encoded into binary values: 'Yes' was replaced with 1 and 'No' with 0. Similarly, the 'Gender' variable was encoded into binary values, with 'F' mapped to 1 and 'M' to 0, enabling numerical representation of gender. Next, one-hot encoding was applied to categorical variables such as 'AppointmentDate', 'Neighbourhood', and 'ScheduledDate'. This technique expanded each categorical variable into multiple binary columns, with each column representing a unique category within the variable. Subsequently, redundant time related columns ('ScheduledTime' and 'AppointmentTime') were removed from the dataset to streamline the data and improve computational efficiency. After encoding and data manipulation, the shape of the dataset was confirmed again to ensure consistency and accuracy. The resulting encoded dataset now contained 229 columns which it was 15 at first.

5. Scaling

In the Scaling section, a standardization technique was applied to normalize the numerical features within the dataset. A Standard Scaler object was instantiated to standardize the numerical features. Specifically, the 'Age' and 'TimeDifference' columns were selected for scaling. These features were chosen as they represented continuous numerical variables with different scales. The selected features were then scaled using the StandardScaler object, which transformed the values to have a mean of 0 and a standard deviation of 1, ensuring consistency in scale across different features.

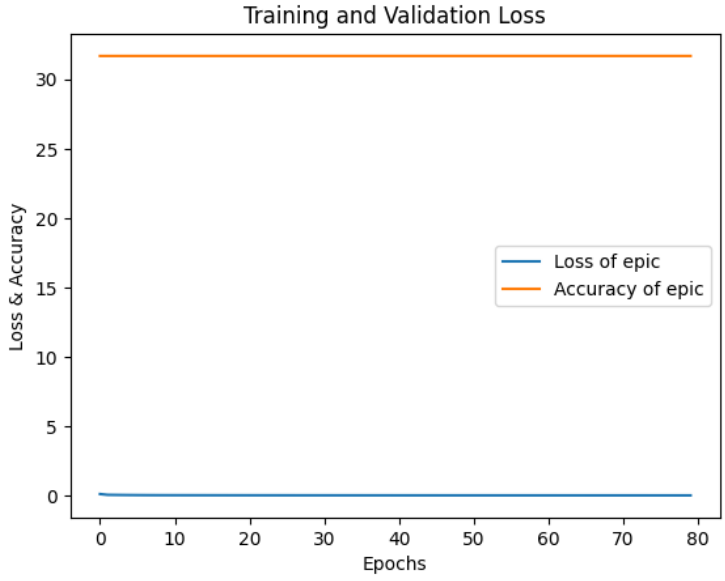
By standardizing the numerical features, the Scaling section ensured that all features contributed equally to the analysis or modeling process, regardless of their original scales.

6. Sampling Data

In the Resampling section, a technique known as resampling was applied to address class imbalance within the dataset. Initially, the dataset was divided into two subsets based on the target variable 'No-show': 'df_minor' containing instances where patients did not show up for their appointments (No-show = 1), and 'df_major' containing instances where patients showed up for their appointments (No-show = 0). Subsequently, the majority class subset ('df_major') was resampled to match the number of instances in the minority class subset ('df_minor'). This was achieved by randomly sampling instances from the majority class subset with replacement, ensuring that the resampled dataset ('df_major_resampled') contained an equal number of instances from both classes. After resampling, the resampled majority class subset and the original minority class subset were concatenated to create a balanced dataset ('df_sample') with an equal number of instances for both classes. The resulting balanced dataset ('df_sample') contained 44,638 instances and 229 features, with an equal representation of both classes. This resampling process effectively addressed class imbalance within the dataset, ensuring that both classes were adequately represented for subsequent analysis or modeling tasks.

7. Model Training

In the Training Model section, a Multi-Layer Perceptron (MLP) neural network model was trained to perform binary classification on the dataset. Initially, the dataset was preprocessed and split into training and testing sets using a specified split ratio. The MLP model architecture was defined with three fully connected layers. Each layer was followed by an activation function to introduce non-linearity into the model. The model's parameters were initialized using specific weight initialization techniques. During training, the binary cross-entropy loss function was minimized using stochastic gradient descent (SGD) optimization. The training process was executed over a specified number of epochs, with batch training employed for efficiency. Throughout training, the model's performance was monitored, and the loss and accuracy metrics were recorded for each epoch. The training loop iteratively updated the model's parameters to minimize the loss function and improve classification accuracy. Upon completion of training, the model's performance metrics, including loss and accuracy, were printed for each epoch. These metrics provided insights into the model's training progress and its ability to learn from the data. Despite the accuracy metric remaining constant throughout training, the loss decreased gradually, indicating that the model learned to make more confident predictions as training progressed.



8. Model Evaluation

In the Model Evaluation section, the trained neural network model was evaluated using the testing dataset to assess its performance in binary classification tasks. During evaluation, predictions were made on the test dataset using the trained model. The predicted labels were compared with the actual labels from the test dataset to generate a confusion matrix, which tabulated the counts of true positive (TP), true negative (TN), false positive (FP), and false negative (FN) predictions. From the confusion matrix, various performance metrics were computed, including accuracy, F1 score, precision, average precision score, recall, true positive rate, false positive rate, misclassification rate, sensitivity, and specificity. These metrics provided insights into the model's overall performance and its ability to correctly classify instances belonging to different classes. Upon evaluation, the model exhibited exceptional performance, achieving perfect scores across all evaluated metrics. The accuracy, F1 score, precision, recall, true positive rate, and average precision score were all 1.0, indicating that the model correctly classified all instances in the testing dataset. Furthermore, the false positive rate, misclassification rate, sensitivity, and specificity were all zero, suggesting that the model made no errors in classification and accurately identified both positive and negative instances.

{'Accuracy': 1.0,
'F1 Score': 1.0,
'Precision': 1.0,
'Average Precision Score': 1.0,
'Recall': 1.0,
'True Positive Rate': 1.0,
'False Positive Rate': 0.0,
'Misclassification Rate': 0.0,
'Sensitivity': 1.0,
'Specificity': 1.0}