|  |
| --- |
| IDS Attacks |
|  |

|  |  |
| --- | --- |
| Team 3 | |
| Name | **ID** |
| Abdallah Hegazy | 2205228 |
| Ziad Mahmoud | 2205021 |
| Maryam Waheed | 2205154 |

Overview Of IDS Attacks :

IDS attacks target intrusion detection systems to bypass or disable them. IDS is designed to monitor network traffic or host activities for suspicious behavior.

In our dataset there is 4 main categories of attack type which are:

* DOS :

DOS attacks aims to overwhelm target system , network or services to make them unavailable to legitimate users which leads to poor performance and may force system to crash.

* Probe:

Probe attacks involve scanning networks to gather information about active services or vulnerabilities for potential exploitation.

* R2L (Remote to local):

R2L attacks occur when attacker remotely gains unauthorized access to a local system by exploiting vulnerabilities or weak authentication.

* U2R (User to root) :

U2R attacks occur when attacker who has local access to the system gain root or administrative permissions or privileges by exploiting vulnerabilities or misconfigurations.

Explanation of dataset preprocessing and model applied :



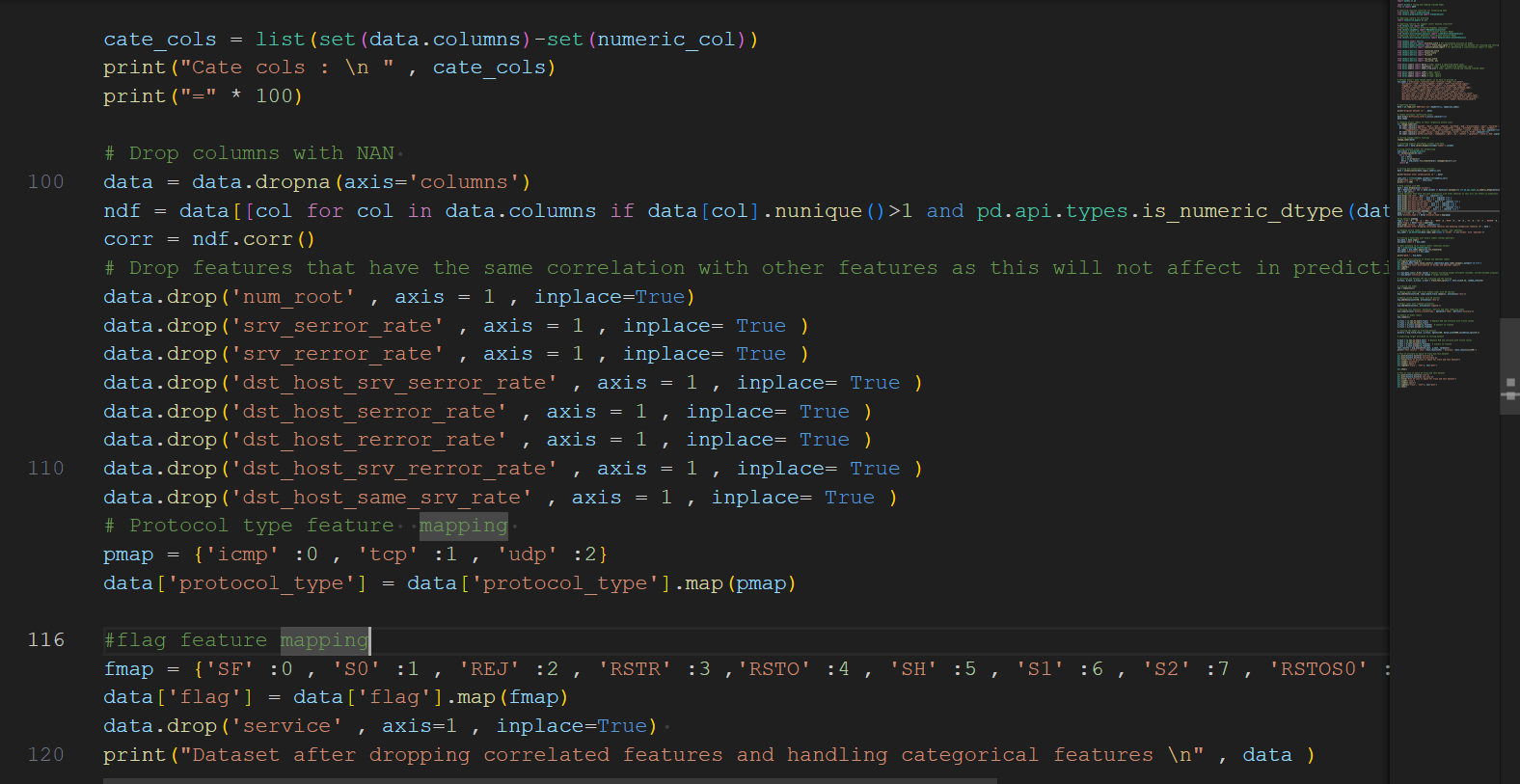
Part 1 ( Loading data set ) :

* Dataset has no column names so we will define names in list called col\_names.
* Reading data from csv file and add column names defined to the data then print data.
* Original data without any feature dropping or preprocessing has 43 columns and 125973 rows.
* Drop difficulty level feature from data so number of rows will be 42 instead of 43
* For more information, describe function is used.

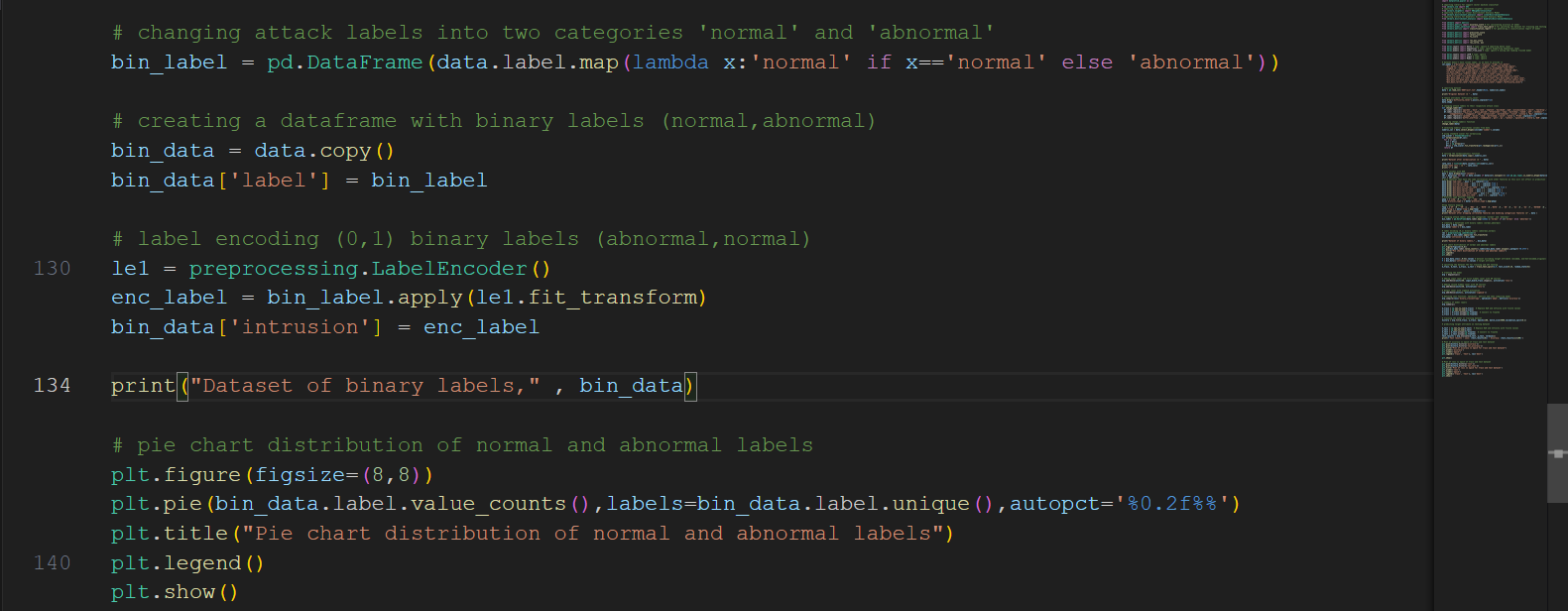


Part 2 (Preprocessing ):

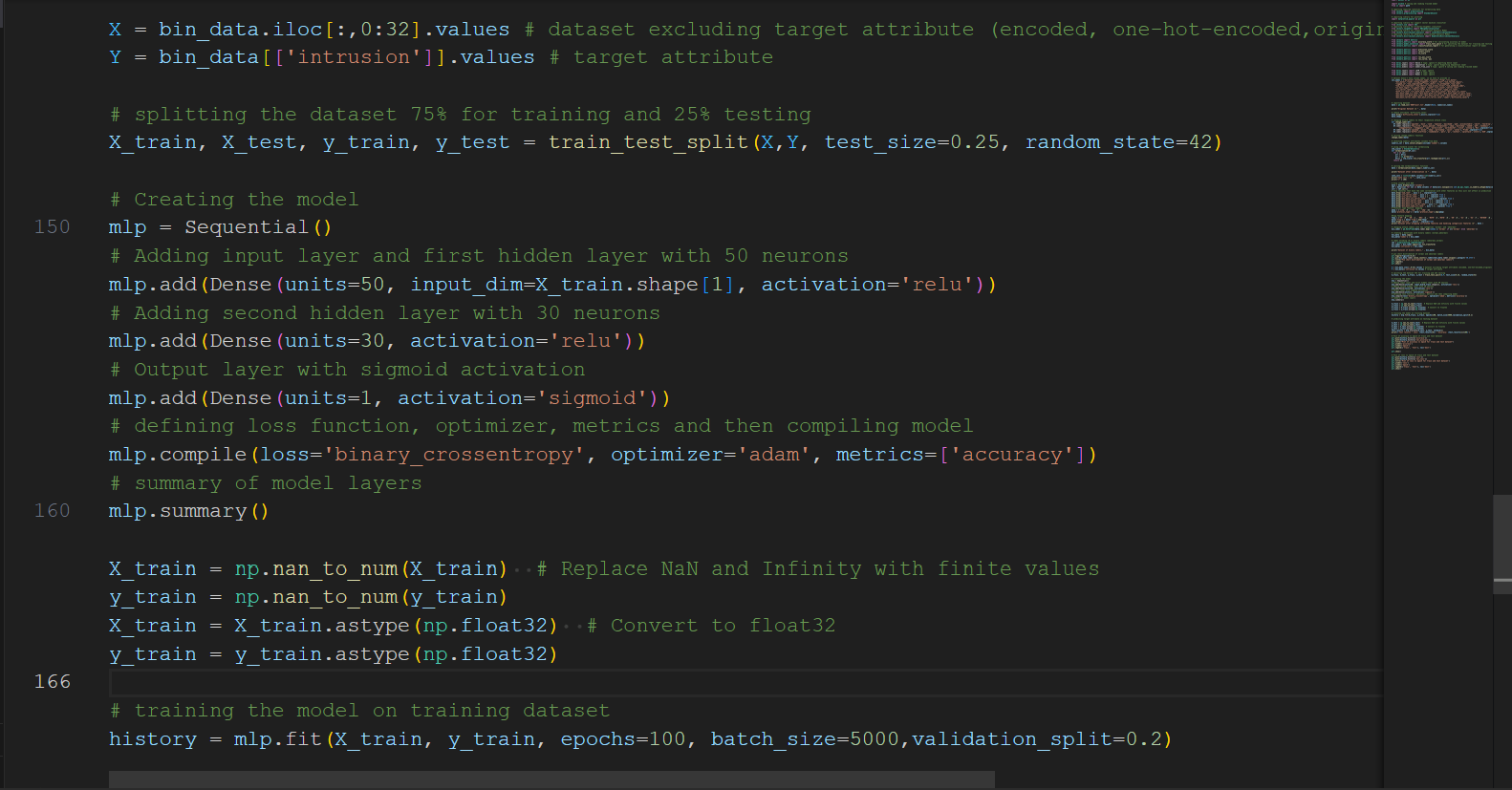
* Maps label values to its attack categories using change.
* Call this function so the label feature values in data will be attack category.
* Fetch features that have numeric values
* Apply data normalization on numerical features only to make them has the same importance.
* Using standard scaler for normalization
* Function normalization is used to normalize each value.
* Then call this function and pass copy of dataset and numeric features to be normalized



* Drop columns with null values.
* Iterate on the data columns and check columns that have more than one unique value as unique value would not be useful in correlation analysis and also check that only numeric columns are included.
* Compute correlation to drop features that have the same correlation as this would not affect on the label.
* As we fetch numeric features from data so to know categorical features we can minus all dataset from numeric features.
* To handle categorical features assign different number for unique values in protocol type , flag and drop service feature.
* Now dataset has 33 features including label.



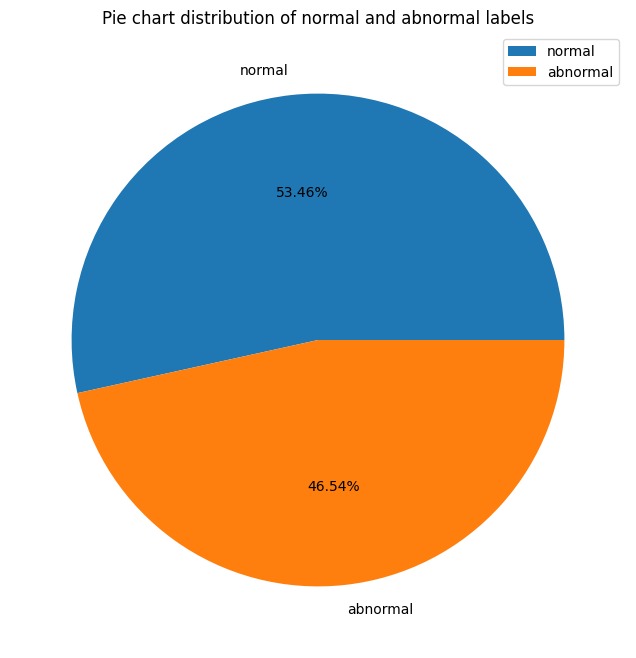
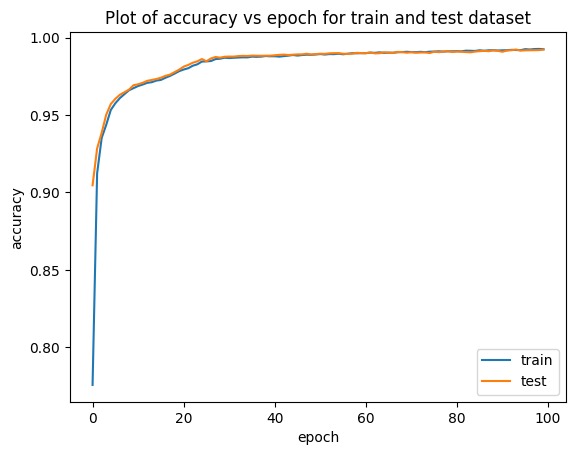
* Label feature has 4 unique values we will maps them into 2 category (normal & abnormal) using lambda function
* Using label encoder to make binary label ( 1 for normal and 0 for abnormal)
* Binary label is named intrusion.
* Now number of features = 34
* Plot pie chart to show the distribution of normal and abnormal.



Part 3 (Data Splitting and modeling ):

* Define X to have all input features which are 32 features (label and binary label “intrusion”) are not included.
* Define Y to have the final label which is intrusion features that determine normal and abnormal.
* Split data 75% training and 25% testing.
* Apply MLP neural network model.
* Define the first layer (input layer) takes 32 input features, apply RELU as activation function and perform 50 neurons
* Define second layer (Hidden layer) that performs 30 neurons and take its inputs from the output of the previous layer and also apply RELU activation function.
* Define the last layer (Output layer) that perform 1 neuron and apply sigmoid activation function
* Define binary cross entropy as loss function && adam optimization accuracy metric to evaluate model
* Training the model into the training & testing set by split dataset into 5000 batches and train data into 100 epochs..

Plots :

========================================================================