MINISTRY OF EDUCATION AND SCIENCE OF UKRAINE

NATIONAL TECHNICAL UNIVERSITY OF UKRAINE

" IHORY SIKORSKY KYIV POLYTECHNIC INSTITUTE"

Artem Volokyta

Software Security

Lab Work 1

kulubecioglu Mehmet

Class Number 12

IM-14 FIOT

Kyiv

IHORY SIKORSKY KYIV POLYTECHNIC INSTITUTE

2024

Intrusion Detection System with Machine Learning

Introduction

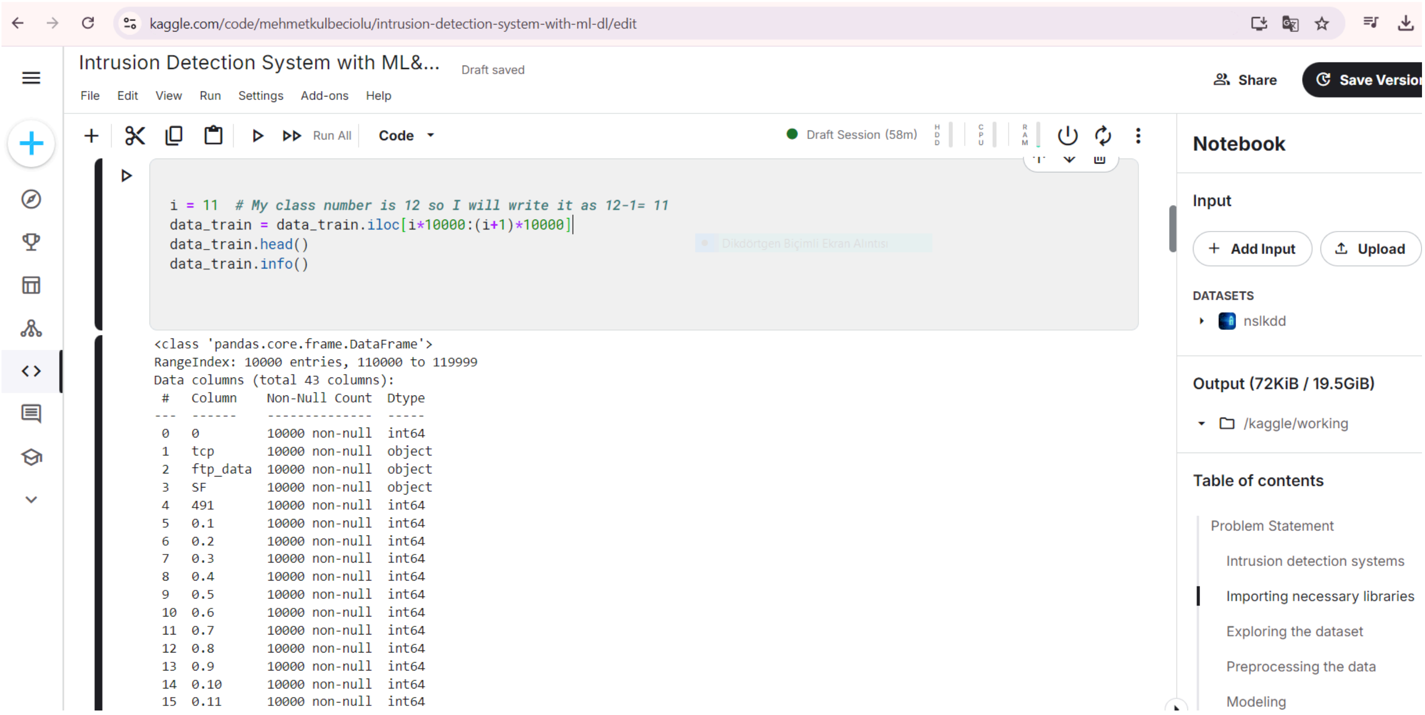
This report covers the work done to develop an Intrusion Detection System (IDS) using the NSL-KDD dataset available on the Kaggle platform. In this process, machine learning methods were employed to analyze and classify the data.

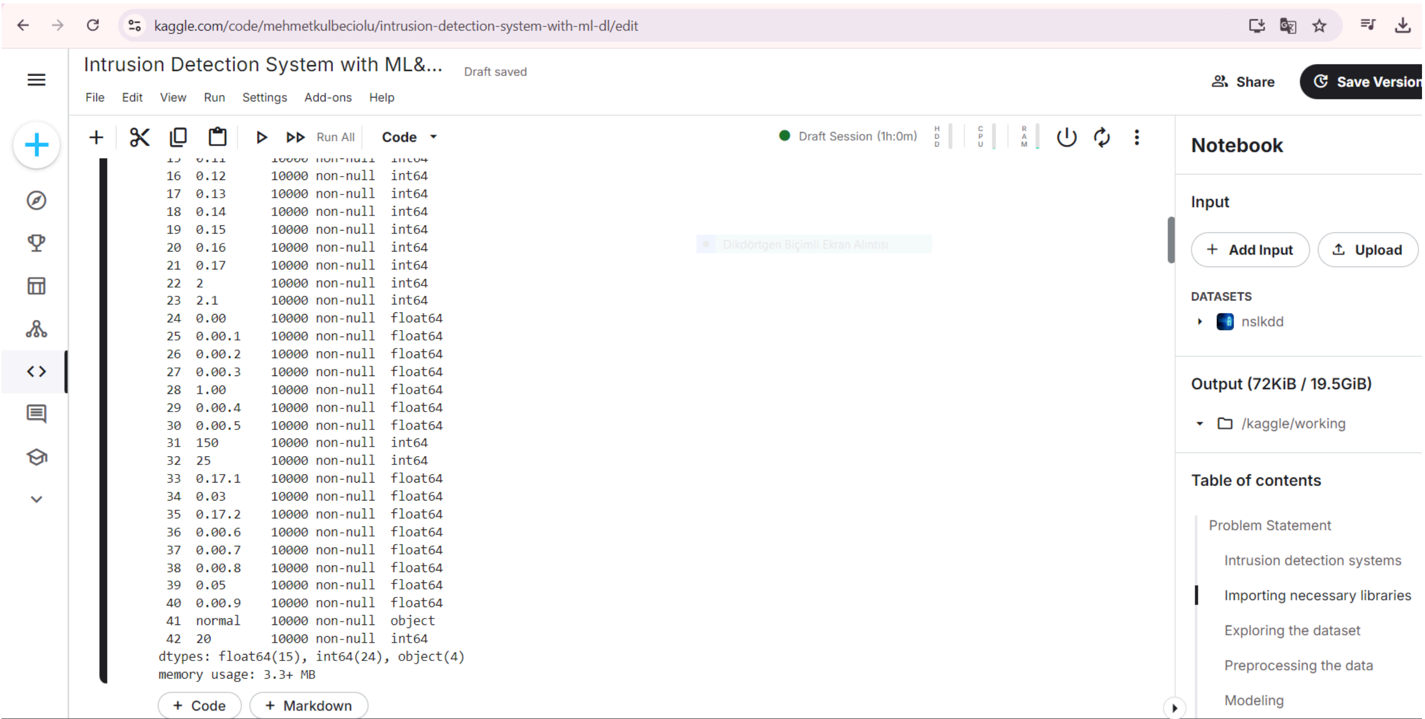
Step 1: Finding the NSL-KDD Dataset and Notebook

First, I logged into my Kaggle account and navigated to the NSL-KDD dataset page. I scrolled down the page to find the 'Most Votes' list, where I located the notebook titled 'Intrusion detection with ML&DL.' I clicked the 'Create Copy & Edit' button in the upper right corner of the notebook to create a copy of it in my Kaggle environment.

Step 2: Selecting a Subset of 10,000 Rows from the Dataset

In the notebook, I proceeded to the 'exploring the dataset' section. Here, I needed to add a new code cell immediately below the `data_train.head()` code cell. To do this, I clicked between the cells and pressed the + Code button on the left. I wrote the following code in the new cell:

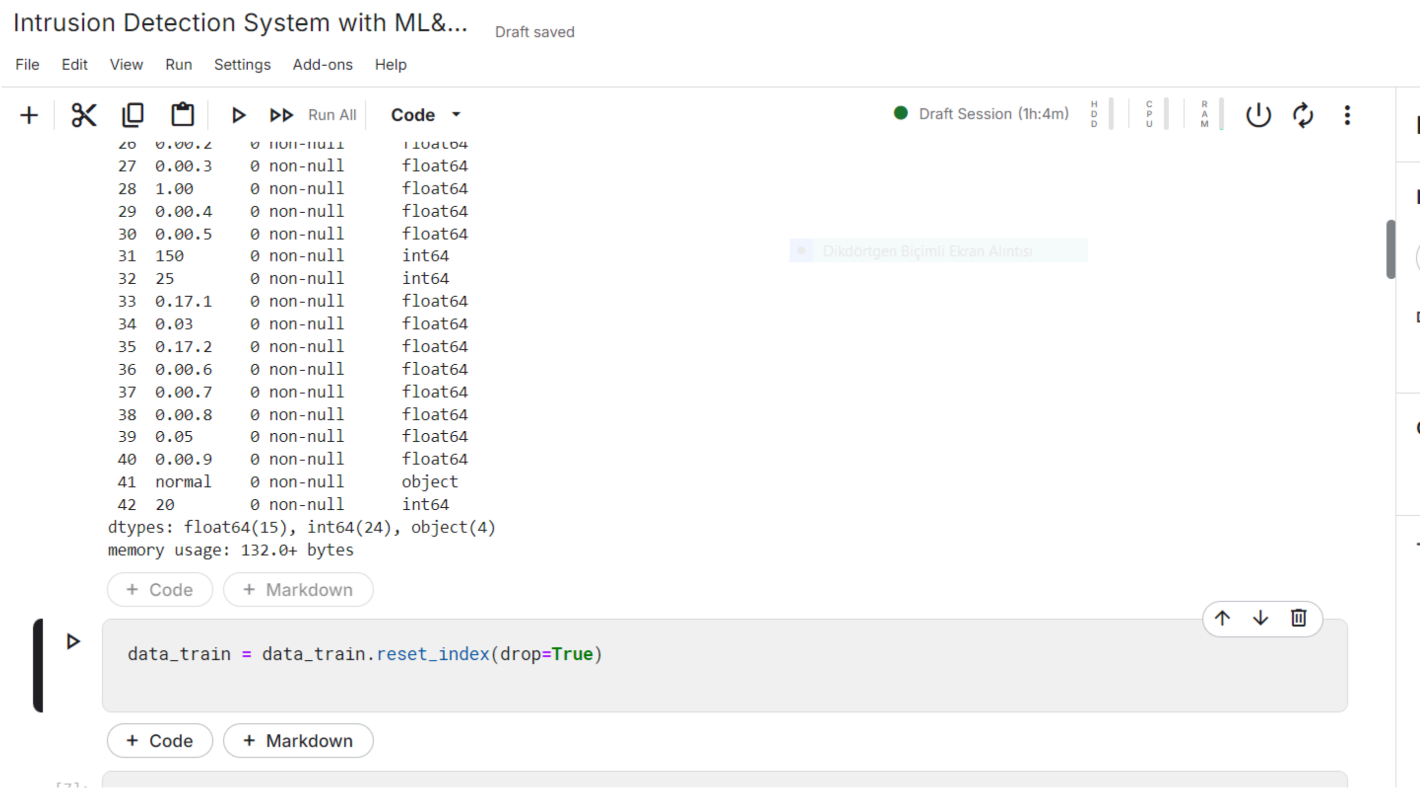




This code allowed me to select a subset of 10,000 rows from the dataset. Next, I ran the `data_train.head()` and `data_train.info()` functions to verify that the data was selected correctly. I saved the outputs of these cells as screenshots.

Resetting the Indices

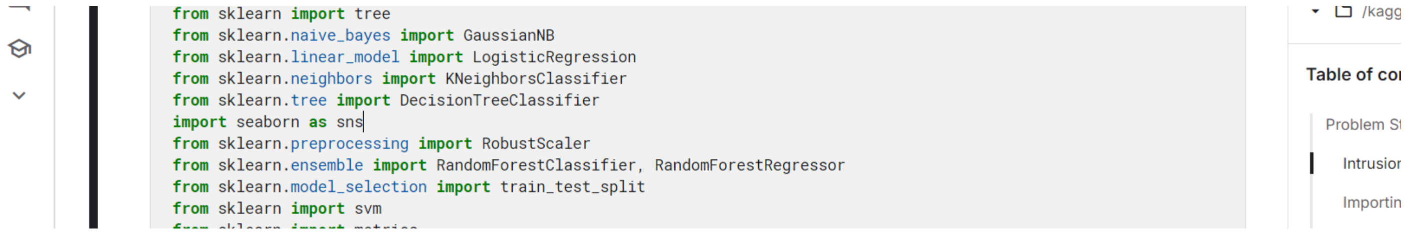
To reset the row indices, I added a code cell and executed the following code:



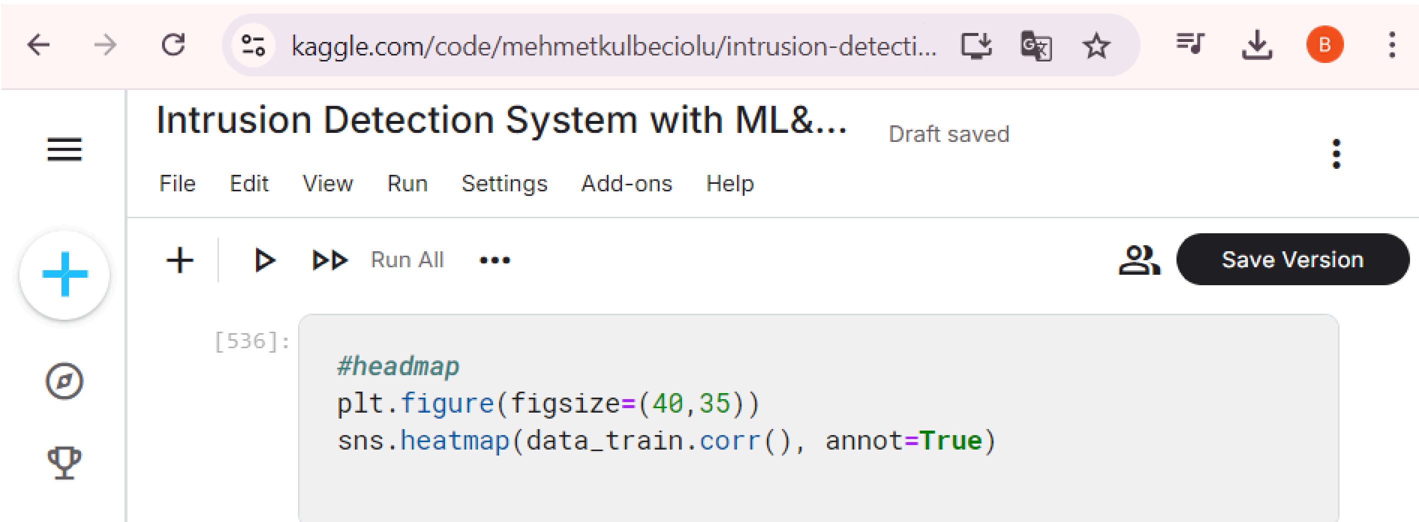
After this operation, I ran the `data_train.head()` cell again to confirm that the indices now started from zero.

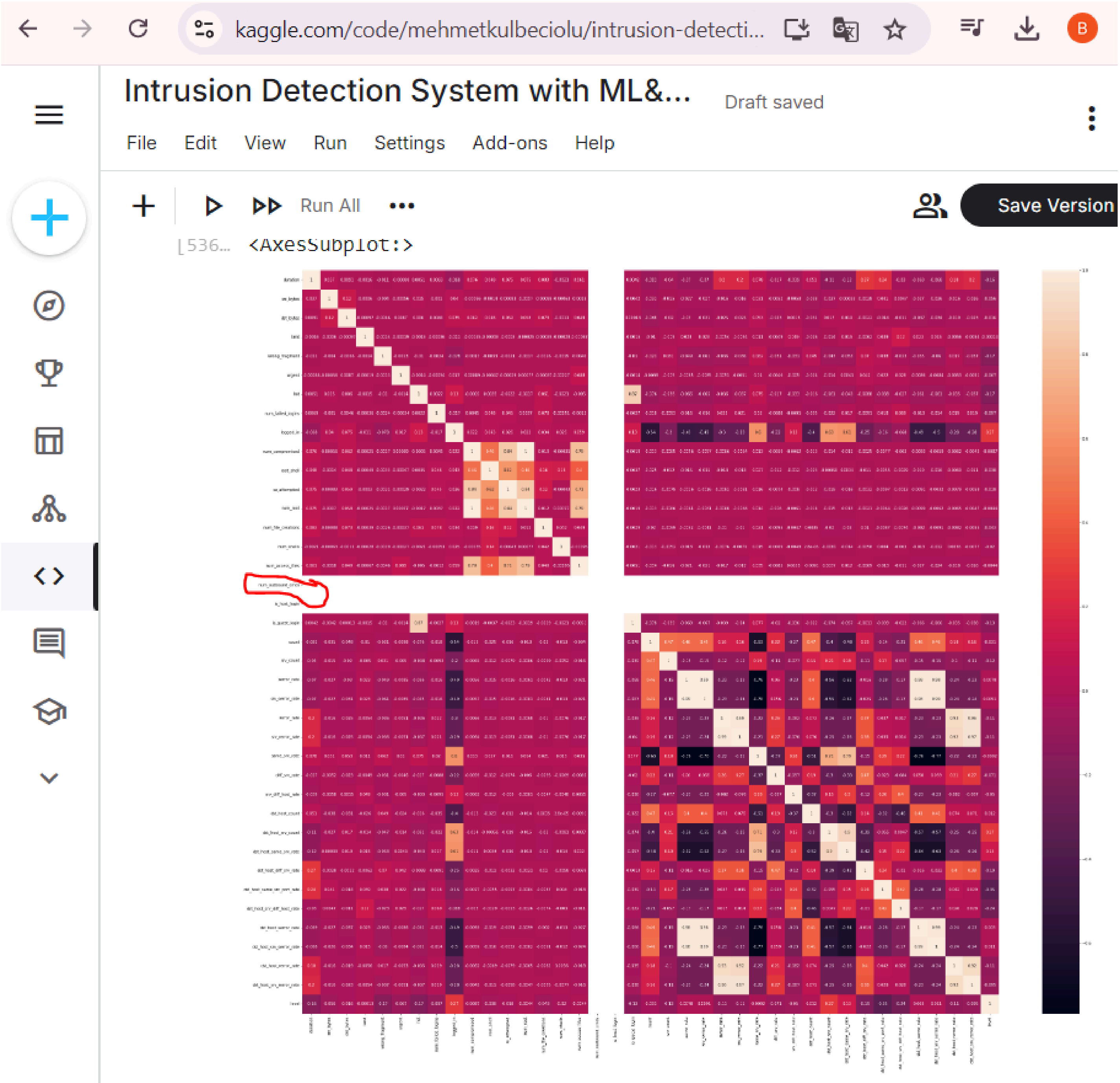
Step 3: Visualizing Data (Creating a Heatmap)

To analyze the data, I needed to use the Seaborn library. I imported Seaborn at the beginning of the notebook:



Then, I added a new cell to create a heatmap to visualize the correlations between the variables in the dataset, writing the following code:





This graph displayed the correlation between the columns in the dataset. Notably, the 'num_outbound_cmds' column appeared empty. If this column did not show any correlation, I decided to remove it from the dataset:

[537]:

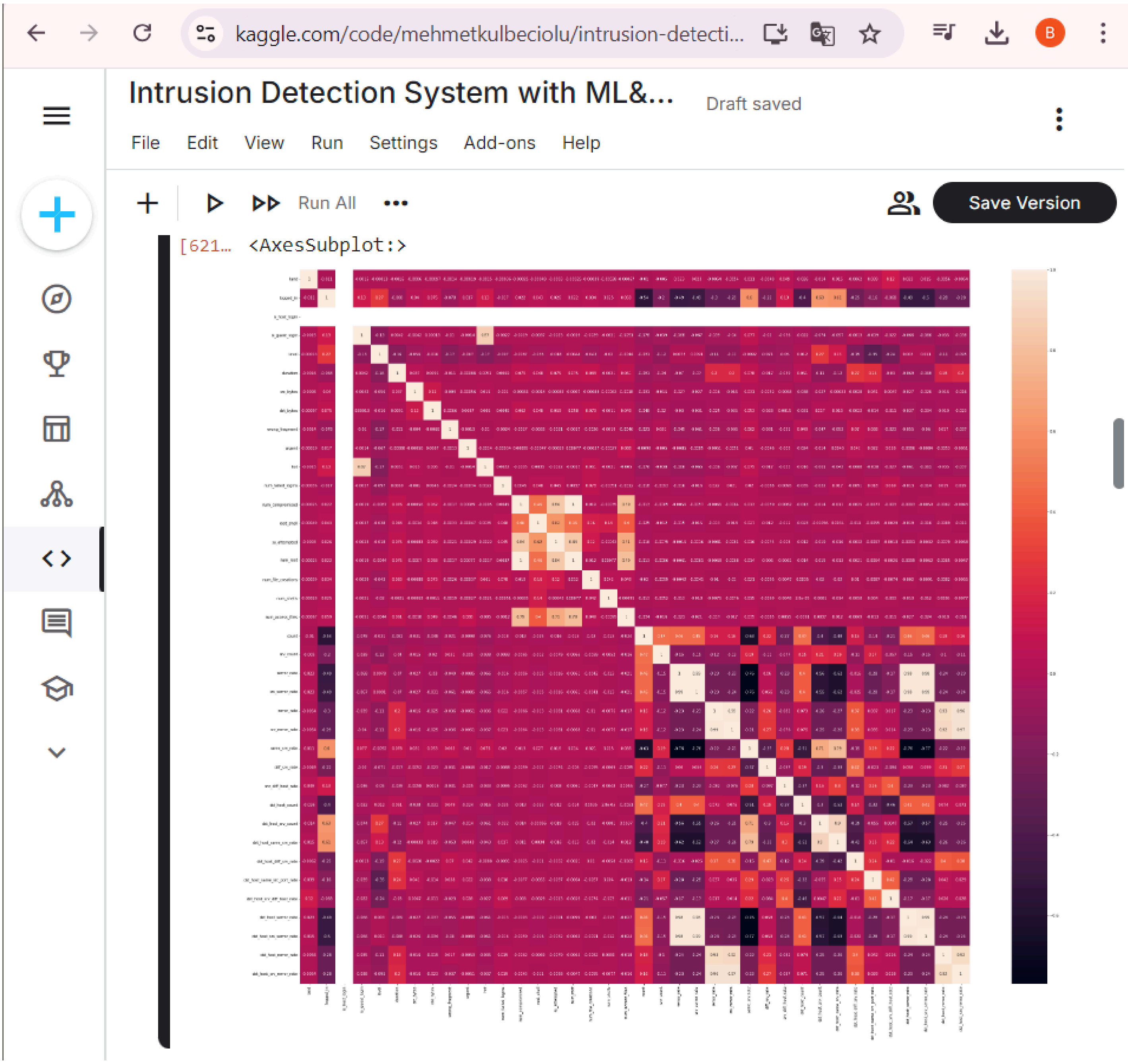
```python
data_train['num_outbound_cmds'].describe()
```

[537…

```
count    10000.0
mean         0.0
std          0.0
min          0.0
25%          0.0
50%          0.0
75%          0.0
max          0.0
Name: num_outbound_cmds, dtype: float64
```

[538]:

```python
data_train.drop('num_outbound_cmds', axis=1, inplace=True)
```



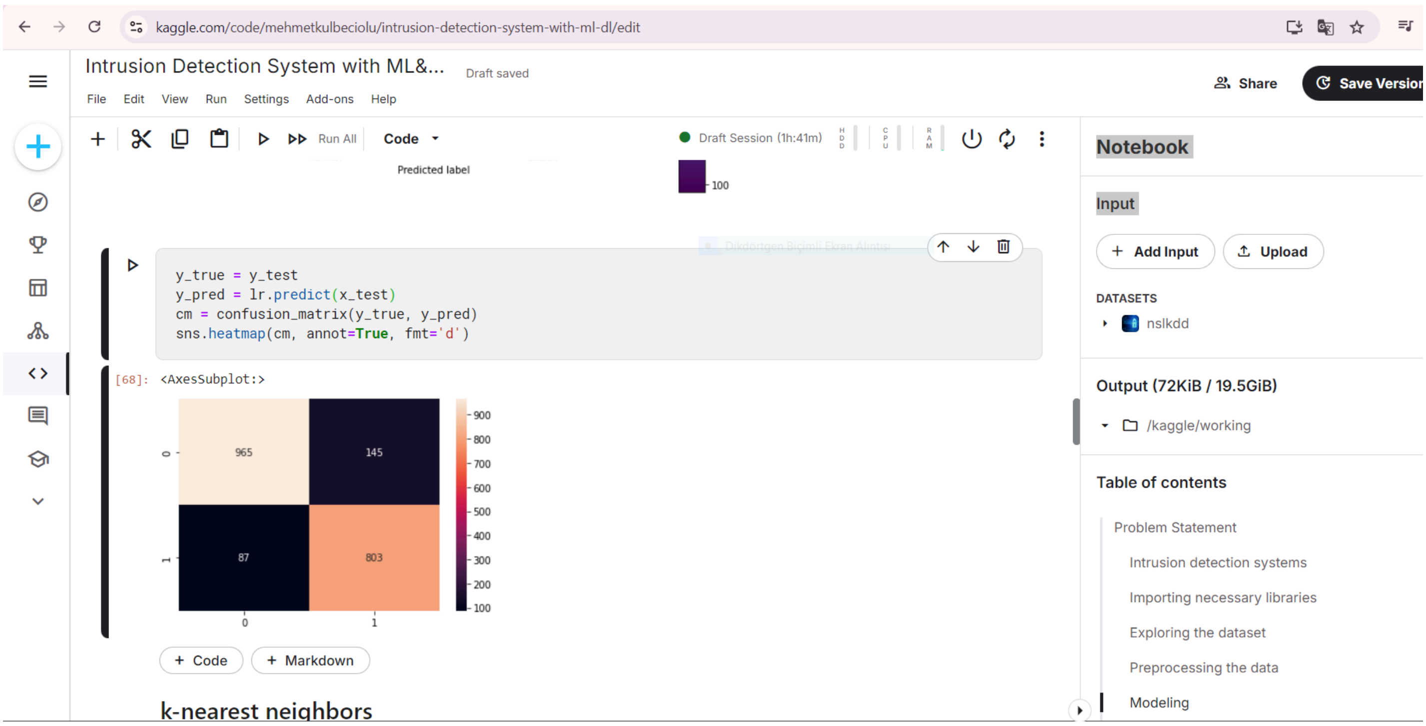
After performing this operation, I ran the heatmap graph again to confirm that the column had been removed.

Step 4: Working with Machine Learning Models

In the subsequent sections of the notebook, I began executing machine learning models. I started with the "logical regression" section. I ran the prepared cells one by one to obtain the results of various models, examining the performance of different algorithms.

In this notebook, the types of attacks in the dataset were transformed into a binary classification problem. This means there were two main categories: attacks and normal behaviors. The model outputs were classified as 0 (no attack) or 1 (attack).

I visualized the results by creating a confusion matrix in one of the cells. This matrix displayed the distribution of correct and incorrect predictions across different classes. My sample code was as follows:



The matrix had four sections:

- Top Left: True Negatives (correctly predicted non-attack cases)

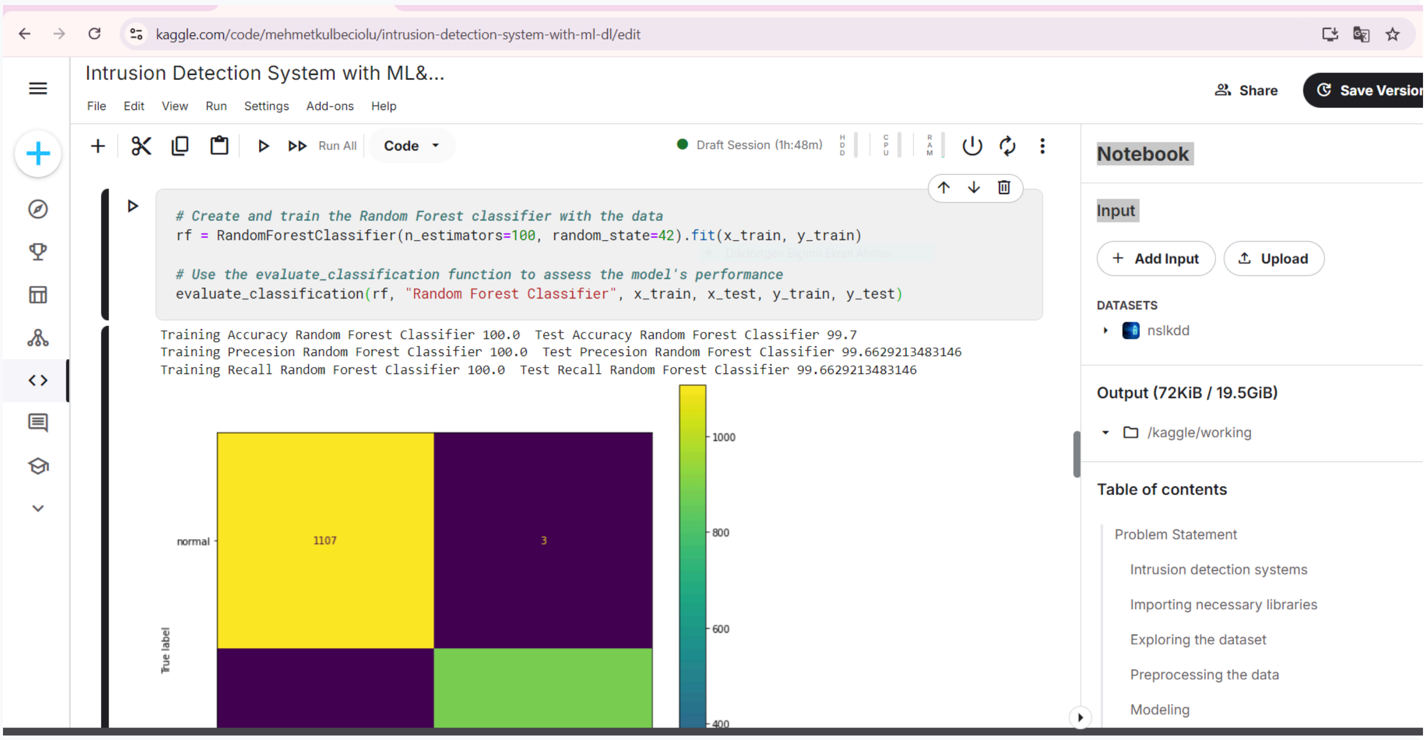
- Bottom Right: True Positives (correctly predicted attacks)

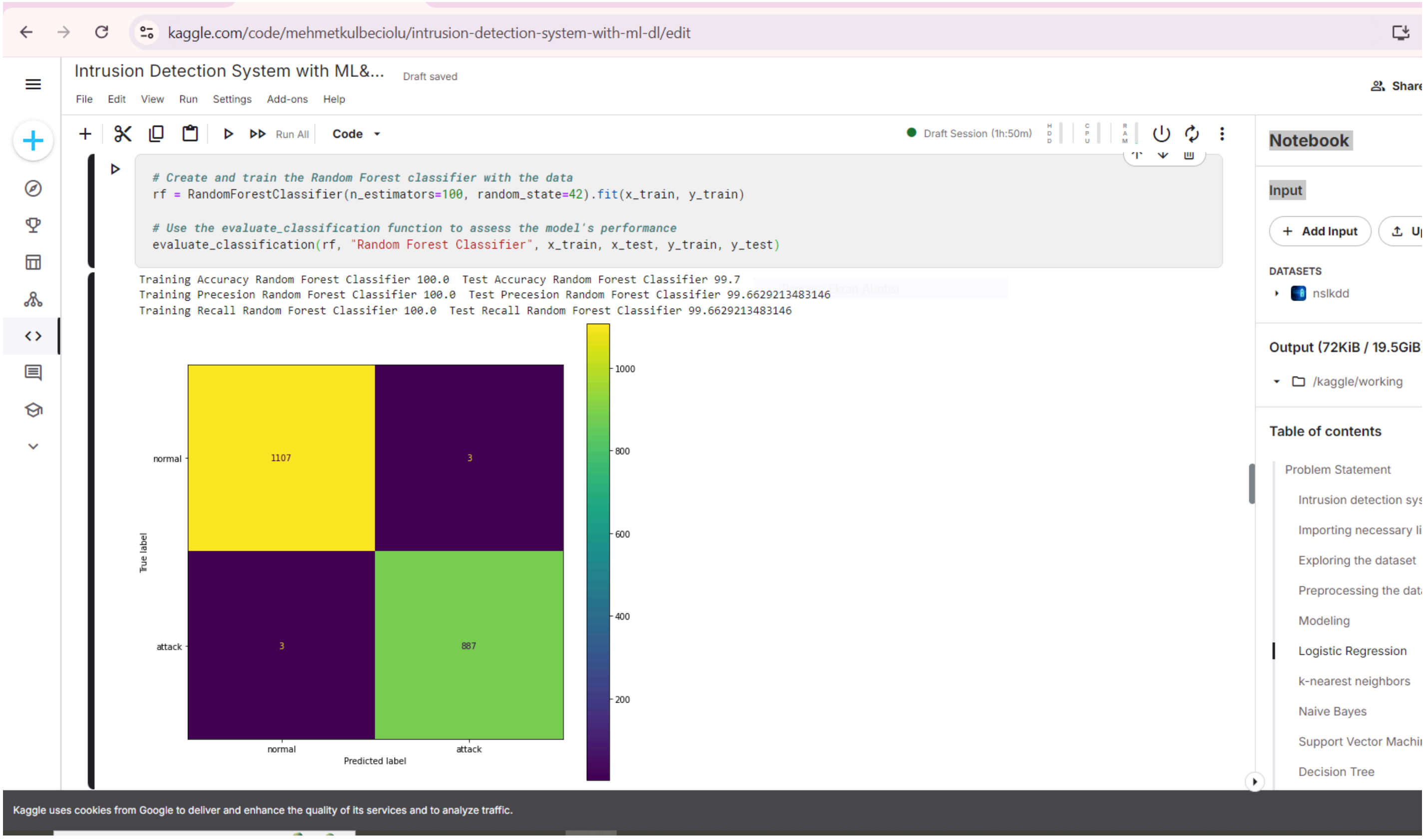
- Bottom Left: False Negatives (attacks incorrectly predicted as normal)

- Top Right: False Positives (normals incorrectly predicted as attacks)

Step 5: Trying Additional Models (Optional)

I also tried out other models from the sklearn library in the notebook. I examined the results of these models and compared how different algorithms performed. For example, I used the Random Forest algorithm as follows:





Conclusion

This laboratory work provided significant experience in developing an intrusion detection system using machine learning techniques with the NSL-KDD dataset. The results obtained will contribute to the development of a more reliable IDS by comparing the performance of different algorithms.