DAC-01-2581\_PRS2023\_ITS

**DIPONEGORO UNIVERSITY**

**DAC-01-0017**

**VINCITORE**

**DAC 2023**

**DAC 2023**

**DAC 2023**

**CHAPTER I: Introduction**

In the swiftly evolving landscape of the modern era, our world is characterized by an unprecedented interplay of technology, communication, and connectivity. These advancement of telecommunications infrastructure has become a cornerstone of progress for nations worldwide. As the world continues its rapid digital transformation, the importance of a robust telecommunications infrastructure cannot be overstated. Indonesia, with its vast expanse and diverse population, heavily relies on telecommunications infrastructure to foster economic development, facilitate education, enhance healthcare services, and enable efficient public administration. The advancement of telecommunications in Indonesia stands as a cornerstone of the nation's growth and prosperity, positioning it as a global player, attracting foreign investments, and facilitating international trade. However, as the Indonesian telecommunications ecosystem matures, the need to address vulnerabilities and protect these critical networks becomes increasingly pressing.

With the proliferation of digital technologies, network attacks have emerged as a genuine threat to the nation's prosperity. According to the Indonesian National Cyber and Crypto Agency (BSSN), incidents of cyberattacks in Indonesia rose by 78% in 2022 alone, with a significant uptick in targeting critical infrastructure. These attacks range from Distributed Denial of Service (DDoS) assaults to ransomware infiltrations, causing financial losses in the billions. Such attacks not only disrupt essential services, but also jeopardize the privacy and security of individuals and organizations alike. The interplay between the rapid expansion of Indonesia's telecommunications infrastructure and the increasing sophistication of network attacks underscores the urgency of robust defense mechanisms.

To defend against the growing menace of network attacks, we have developed software designed to proactively detect and thwart various types of network threats by analyzing the distinctive characteristics of network traffic data using the XGBoost algorithm. In this endeavor, XGBoost has the capability to handle large-scale data, coupled with its efficiency and accuracy, significantly enhances our software's capacity to discern intricate patterns and anomalies indicative of network attacks. In an era where cyberattacks are constantly evolving in sophistication, the combination of software development tailored for network attack prevention and the utilization of XGBoost as the underlying algorithm empowers other organizations to fortify their network defenses, minimize potential risks, and ensure the continuous availability and security of their digital assets.

**CHAPTER II: Theoretical Framework**

XGBoost is a robust and highly effective machine learning algorithm for classification tasks. In XGBoost classification, the primary goal is to assign categorical labels or classes to input data points based on their features. This algorithm leverages an ensemble of decision trees to achieve accurate predictions. Each tree specializes in distinguishing between different classes, and the final prediction is made by aggregating the outputs of these individual trees.

The mathematical foundation of XGBoost's classification objective is an extension of the logistic loss function used in binary classification to accommodate multiple classes (commonly referred to as "softmax" or "multiclass logistic regression"). In the case of multiclass classification with classes, the objective function can be expressed as follows:

Objective

The key components in the formula are given in Table 2.1.

**Table 2.1.** Meaning of the symbols

|  |  |
| --- | --- |
|  | Represents the total number of training examples |
|  | Number of classes in the classification problem |
|  | True label for class of the -th example |
|  | The predicted output for class of the -th example |
| First Term | Calculates the logistic loss for each example and class, comparing the predicted values to the true labels |
| Second Term | Incorporates regularization terms (L1 and L2) to control the complexity of individual trees in the ensemble |

XGBoost's classification objective strives to minimize this function during the training process. By doing so, it adapts to multiclass scenarios, effectively distinguishing between multiple categories and providing accurate classification results. Through its combination of decision trees, regularization techniques, and optimization strategies, XGBoost has proven to be a powerful tool for a wide range of classification tasks, including image recognition, text classification, and more.

**CHAPTER III: Analytical Steps**

In this study, we undertook a comprehensive examination of a dataset encompassing around 110,000 entries with 42 variables. Of these variables, 41 were distinctive attributes, with one variable termed ”type\_of\_attack” serving as the target variable. The primary objective was to uncover relationships among these attributes to forecast the type of network attack.

Upon analyzing the dataset, anomalies, characterized by symbols like ”\*” or ”99999”, were detected in certain variables. To ensure data consistency, these anomalies were replaced with NumPy's ”NaN”, signifying missing values. Instead of adopting traditional methods to handle these anomalies, we employed the XGBoost algorithm. This algorithm is inherently equipped to handle missing data during its training process. It assesses the most suitable split for missing values, choosing the best path in the decision tree. This inherent capability of XGBoost justified its selection for our research.

An essential aspect of our research was the assessment of the distribution of the ”type\_of\_attack” variable. An imbalance was observed: ”normal” attacks represented 53.84%, and ”neptune” attacks comprised 33.03%. Such disparities, if not appropriately addressed, could skew the model's predictive accuracy. While significant imbalances necessitate complex mitigation techniques, subtle imbalances like those in our dataset can be addressed during data stratification, which will be discussed in later sections.

Prior to the modeling phase, the categorical variables in the dataset underwent a transformation process termed one-hot encoding, facilitated by Pandas' get\_dummies function. This method creates new variables for each category, assigning them binary values. This avoids inadvertently establishing ordinal relationships, an issue often linked with label encoding. Given its ability to retain the categorical essence of data without imposing unintended hierarchies, one-hot encoding was deemed most appropriate.

For subsequent analysis, the dataset was divided into training and testing subsets, following a 70:30 proportion. To meticulously manage the observed imbalances in the target variable, the 'stratify' function was employed during this division. This deliberate decision ensured both subsets reflected the distribution observed in the original dataset, paving the way for robust and unbiased predictive modeling.

Following the preliminary data processing, we turned our attention to the XGBoost framework, specifically leveraging the XGBoost classifier for our training endeavors. For this initial phase, a set of standard hyperparameters was employed, sidestepping a blind pursuit of cross-validation with Scikit-learn's GridSearchCV. Our intent was to first assess the primary results, a detailed discussion of which is slated for the subsequent chapter. This strategy was informed by a simple premise: if the initial results prove satisfactory, the exhaustive, computationally-intensive process of GridSearch might be rendered unnecessary. Moreover, such deep dives could elevate the risk of model overfitting. Once the XGBoost model was diligently trained, our next objective was to deploy it for predictions on the testing dataset, setting the stage for a comprehensive evaluation.

**CHAPTER IV: Analysis of Results**

As we delved into our analysis, our initial step was to assess the performance of our model. A cornerstone metric in the evaluation of classification models is accuracy. This measure gauges the overall performance by indicating the proportion of correctly predicted instances. Our model achieved an impressive accuracy score of 99.64%, signifying its exceptional ability to predict the type of network attack.

However, accuracy alone might not give a holistic view, especially when dealing with imbalanced datasets. This led us to explore the classification report, presented in Figure 4.1, which is a comprehensive table offering precision, recall, and the F1-score for each class. Precision calculates the ratio of correctly predicted positive observations to the total predicted positives. Recall, also known as Sensitivity or True Positive Rate, quantifies the number of actual positives our model captures. The F1-Score is the harmonic mean of Precision and Recall and gives a better measure when there's an uneven class distribution.

A screenshot of a computer screen

Description automatically generated

**Figure 4.1.** Classification Report

For instance, focusing on the "ipsweep" attack category, our model achieved a precision of 0.92, indicating that 92% of instances predicted as "ipsweep" were actually "ipsweep" attacks. The recall for this class stood at 0.98, meaning the model identified 98% of all actual "ipsweep" instances. These metrics, combined with an F1-Score of 0.95, suggest that even for specific attack types, our model was highly adept at making accurate predictions.

To further visualize the model's prowess, we delved into the confusion matrix plot presented in Figure 4.2.



**Figure 4.2.** Confusion Matrix Plot

The matrix revealed that our model was predominantly accurate across all classes, including the minor ones. For instance, in the context of the "ipsweep" category, out of 976 true instances, our model correctly classified 958, with only a handful being misclassified. The ability of our model to correctly identify instances of even minor classes underscores its capability to handle imbalances adeptly. In essence, while a few misclassifications were observed, the XGBoost model showcased a remarkable capacity to distinguish between various network attacks.

For any model, understanding which features play a pivotal role in predictions is invaluable. Our feature importance analysis, presented in Figure 4.3, illuminated the variables most influential in predicting the type of network attack.

A graph with numbers and text

Description automatically generated

**Figure 4.3.** Feature Importances Plot

Leading the chart was ”diff\_srv\_rate”, with an importance score of 33.18%. This suggests that ”diff\_srv\_rate” alone was responsible for approximately one-third of the decision-making capability of our model. This was followed by attributes such as ”flag\_RSTR” and ”service\_ecr\_i”, each contributing significantly to the model's decisions. These variables, among others, hold the key to understanding the nuances of network attacks.

**CHAPTER V : Conclusion and Recommendation**

In conclusion, our research into network attack prediction utilizing the XGBoost algorithm has provided valuable insights and outcomes:

1. High Detection Accuracy: The XGBoost-based software achieved a notable accuracy of 99.64% in detecting network threats, ensuring reliable protection for telecommunications infrastructure.
2. Robustness Amidst Data Inconsistencies: Despite frequent missing values and imbalances in real-world datasets, XGBoost effectively handles these challenges through its innate data processing capability and stratification techniques, ensuring its applicability in varied conditions.
3. Key Influential Variables: Variables such as "diff\_srv\_rate," "flag\_RSTR," and "service\_ecr\_i" are significant in predicting network attack types, offering insights into critical attack attributes.

Based on our findings and comprehensive analysis, we propose the following recommendations to enhance cybersecurity and bolster the resilience of Indonesia's telecommunications infrastructure:

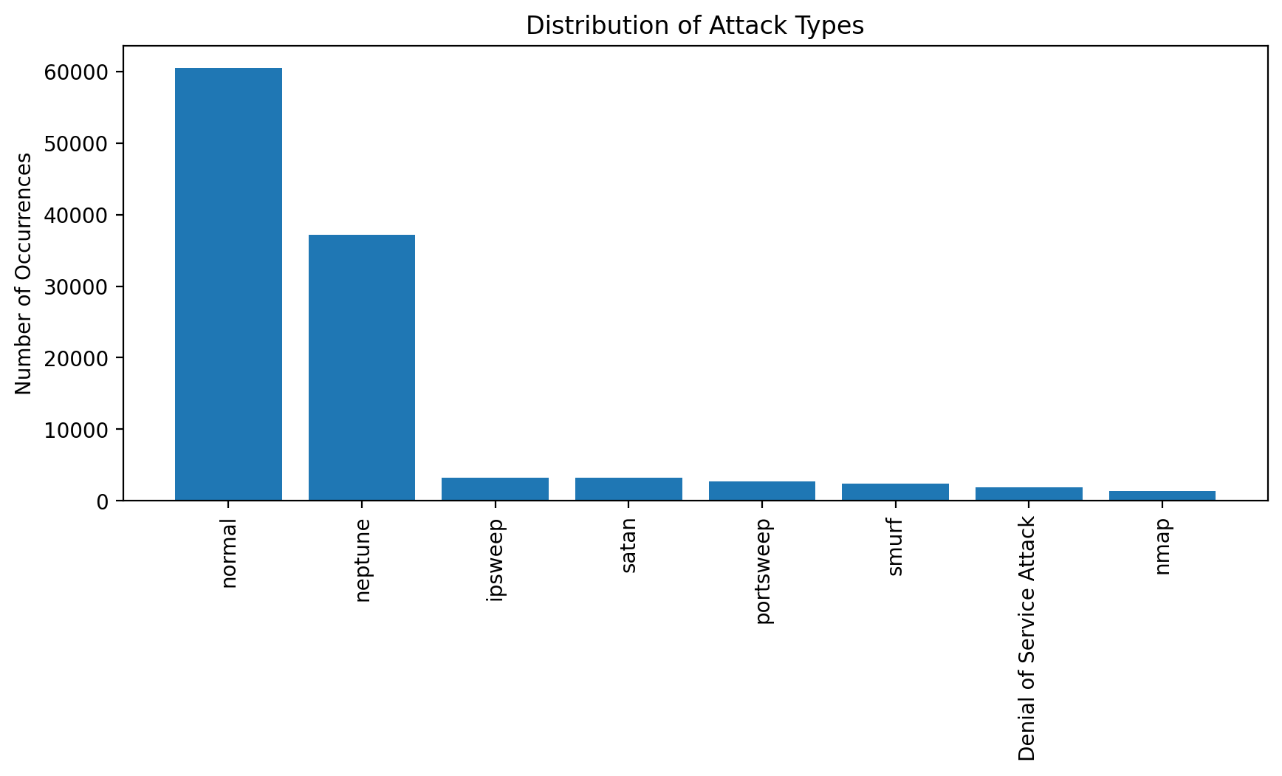
1. Reinforce Cybersecurity Protocols: Organizations should enhance their defense against network attacks through regular cybersecurity training, updating threat intelligence, and fostering collaborations.
2. Model Optimization: Continual refinement of the XGBoost model using updated datasets is crucial. Exploring ensemble methods can further boost accuracy.
3. Incident Preparedness: Institutions should incorporate real-time monitoring systems that leverage the predictive model and establish up-to-date incident response plans for immediate threat mitigation.
4. Regulation Enhancement: Governments should bolster cybersecurity regulations and introduce incentives for compliance, fostering an environment of stringent cybersecurity practices.

# REFERENCES

|  |  |
| --- | --- |
| [1] | H. Ardiyanti, "Cyber Security dan Tantangan Pengembangannya di Indonesia," Politica, vol. 5, no. 1, 2014. |
| [2] | M. Rizki, "Perkembangan Sistem Pertahanan/Keamanan Siber Indonesia dalam Menghadapi Tantangan Perkembangan Teknologi dan Informasi," Politeia, vol. 14, no. 1, 2021. |
| [3] | N. N. P. Pinata, I. M. Sukarsa, and N. K. D. Rusjayanthi, "Prediksi Kecelakaan Lalu Lintas di Bali dengan XGBoost pada Python," Jurnal Ilmiah Merpati, vol. 8, no. 3, Dec. 2020. |
| [4] | S. A. M. Babys, "Ancaman Perang Siber Di Era Digital Dan Solusi Keamanan Nasional Indonesia," Jurnal Oratio Directa, vol. 3, no. 1, 2021. |
| [5] | S. E. H. Yulianti, O. Soesanto, and Y. Sukmawaty, "Penerapan Metode Extreme Gradient Boosting (XGBOOST) pada Klasifikasi Nasabah Kartu Kredit," Journal of Mathematics, vol. 4, no. 1, 2022. |

**ATTACHMENT**

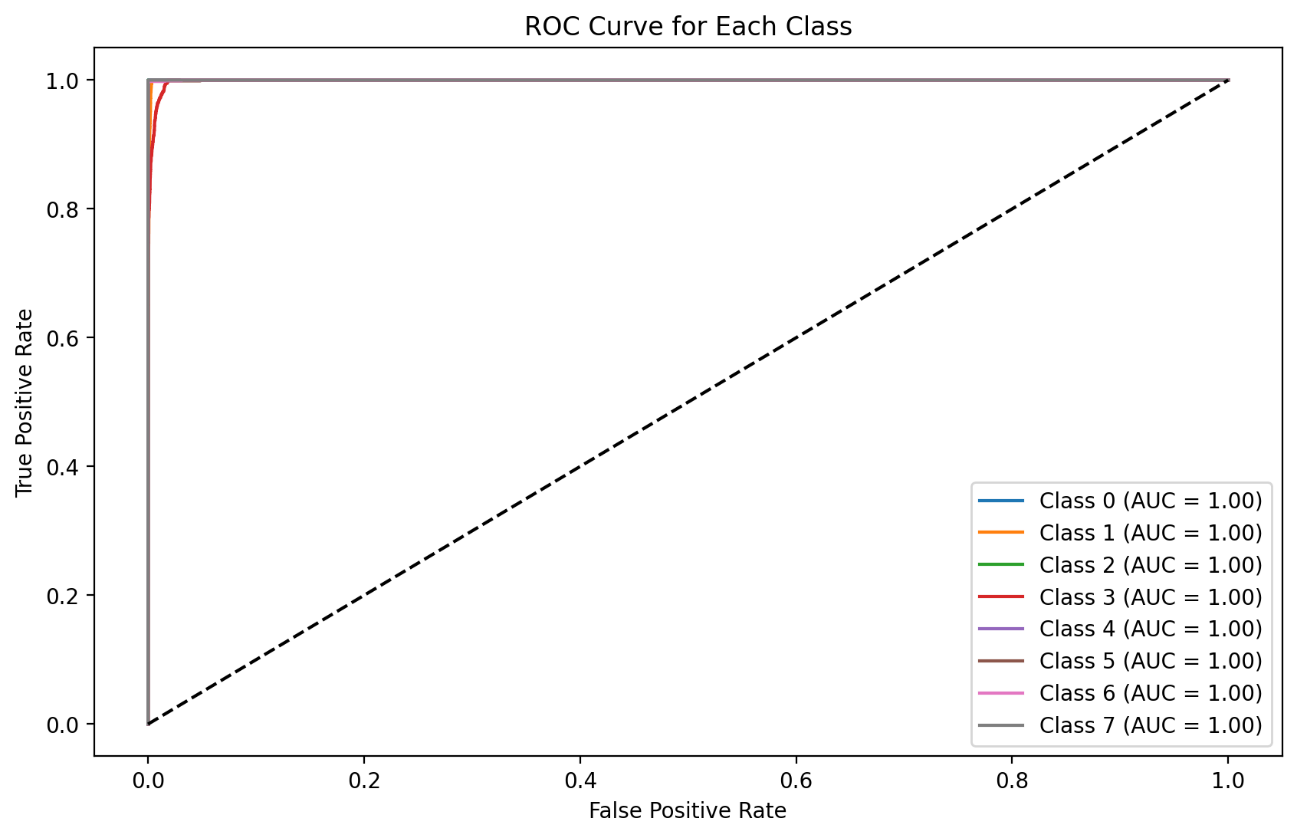
Attachment 1: Distribution of Attack Types



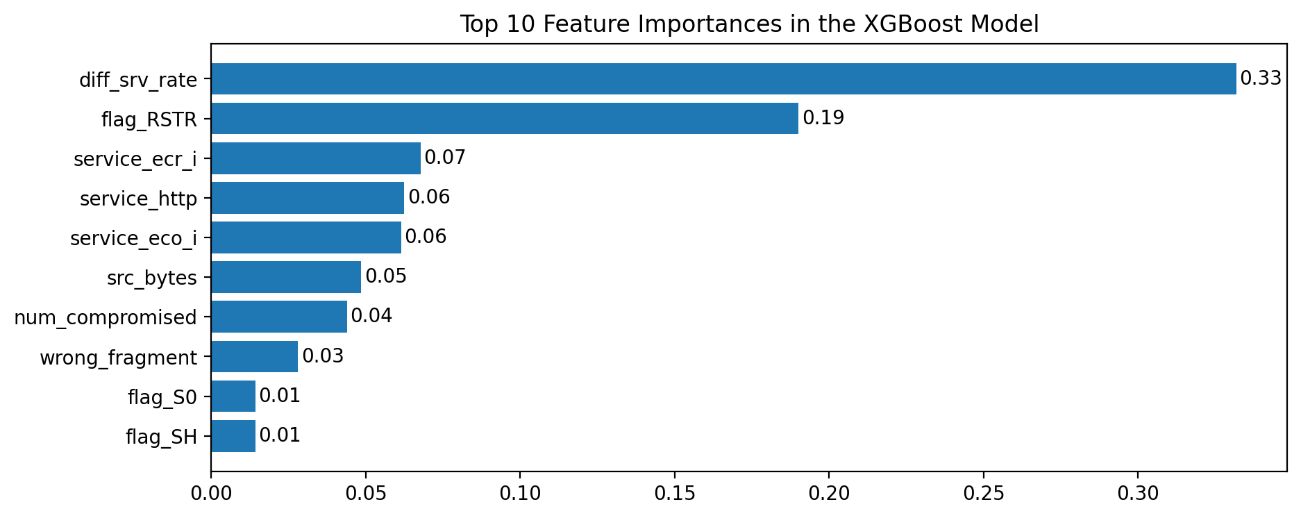
Attachment 2: Confusion Matrix for Attack Type Predictions

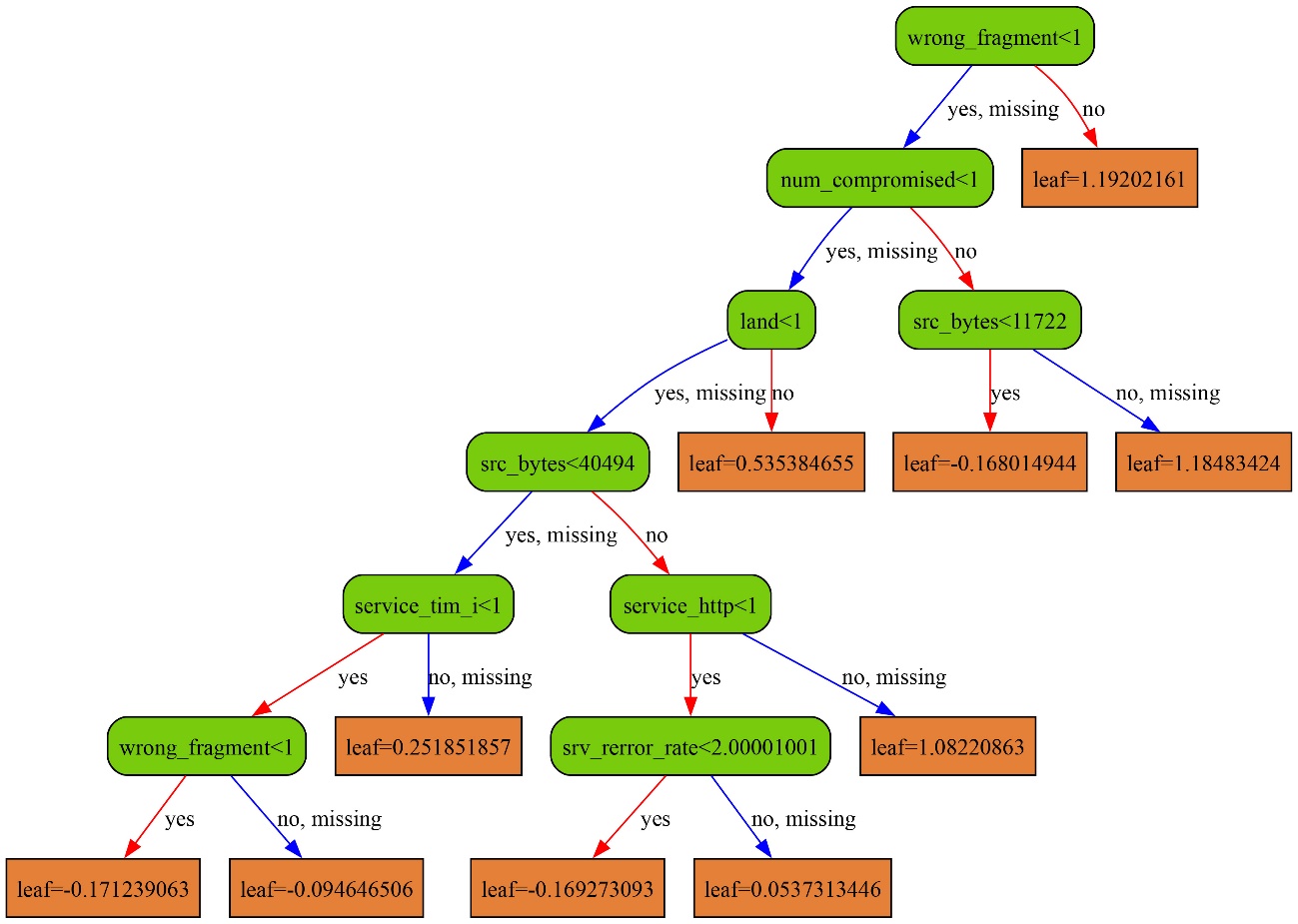


Attachment 3: ROC Curve for Each Class



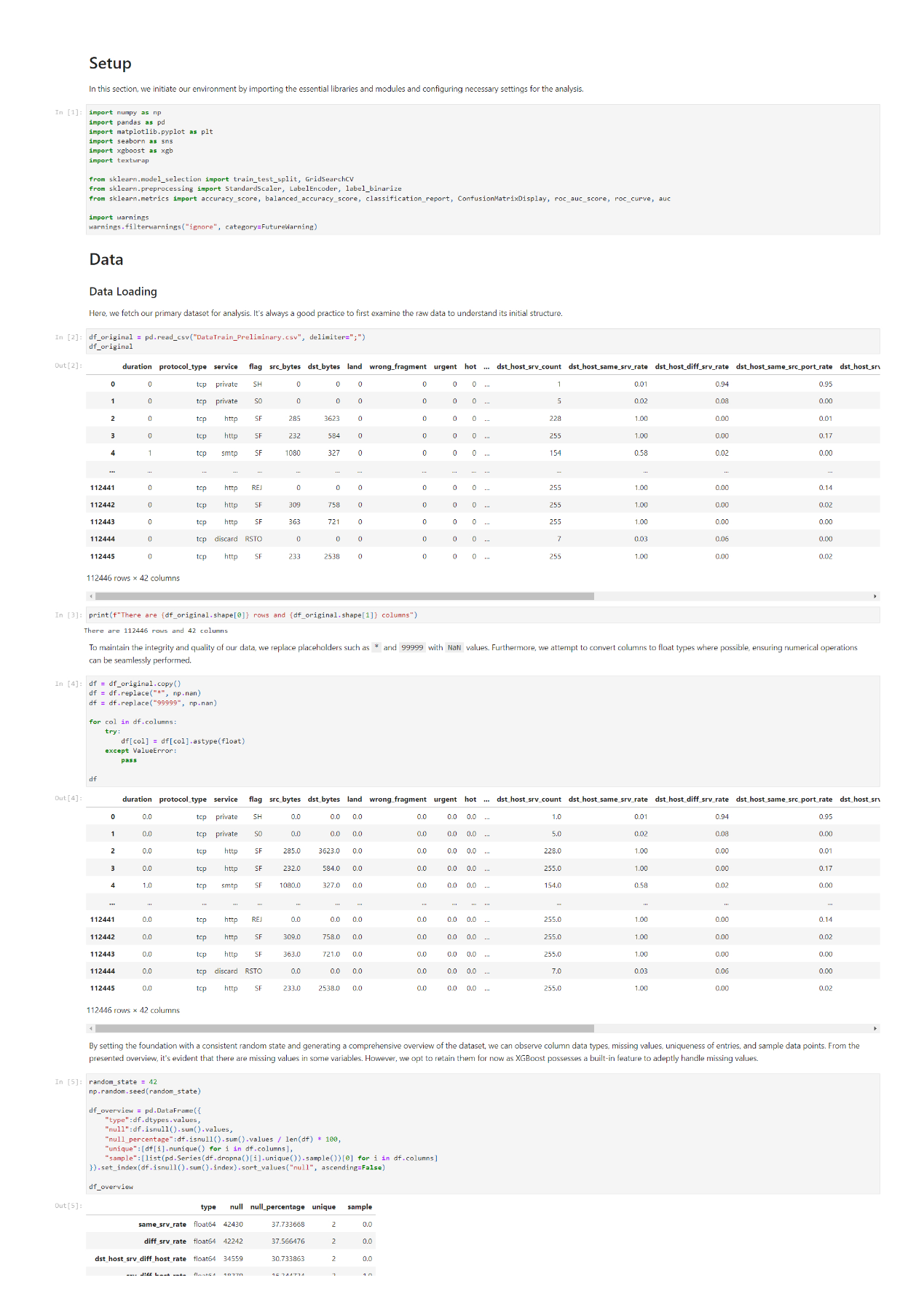
Attachment 4: Top 10 Feature Importances in the XGBoost Model

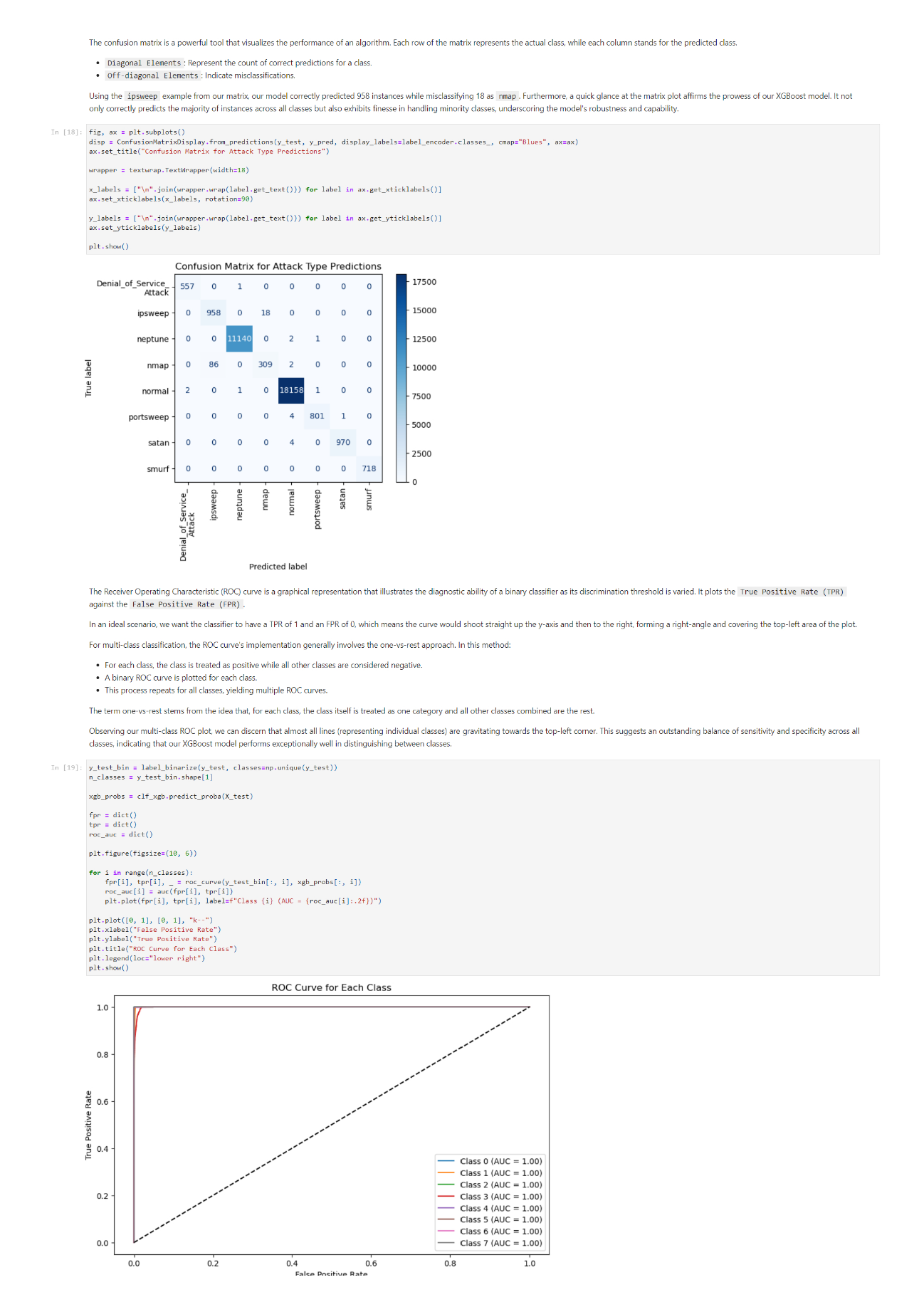
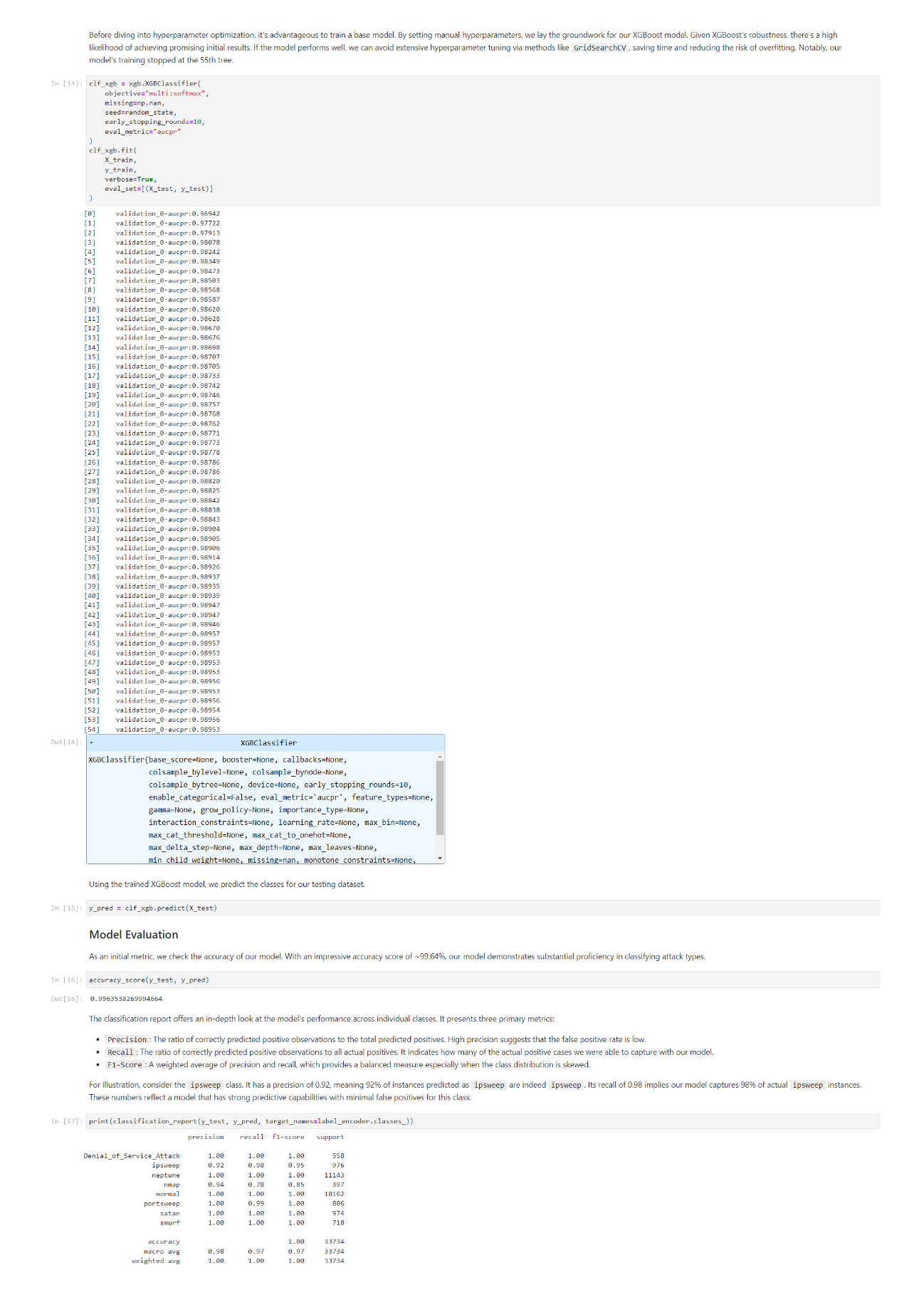
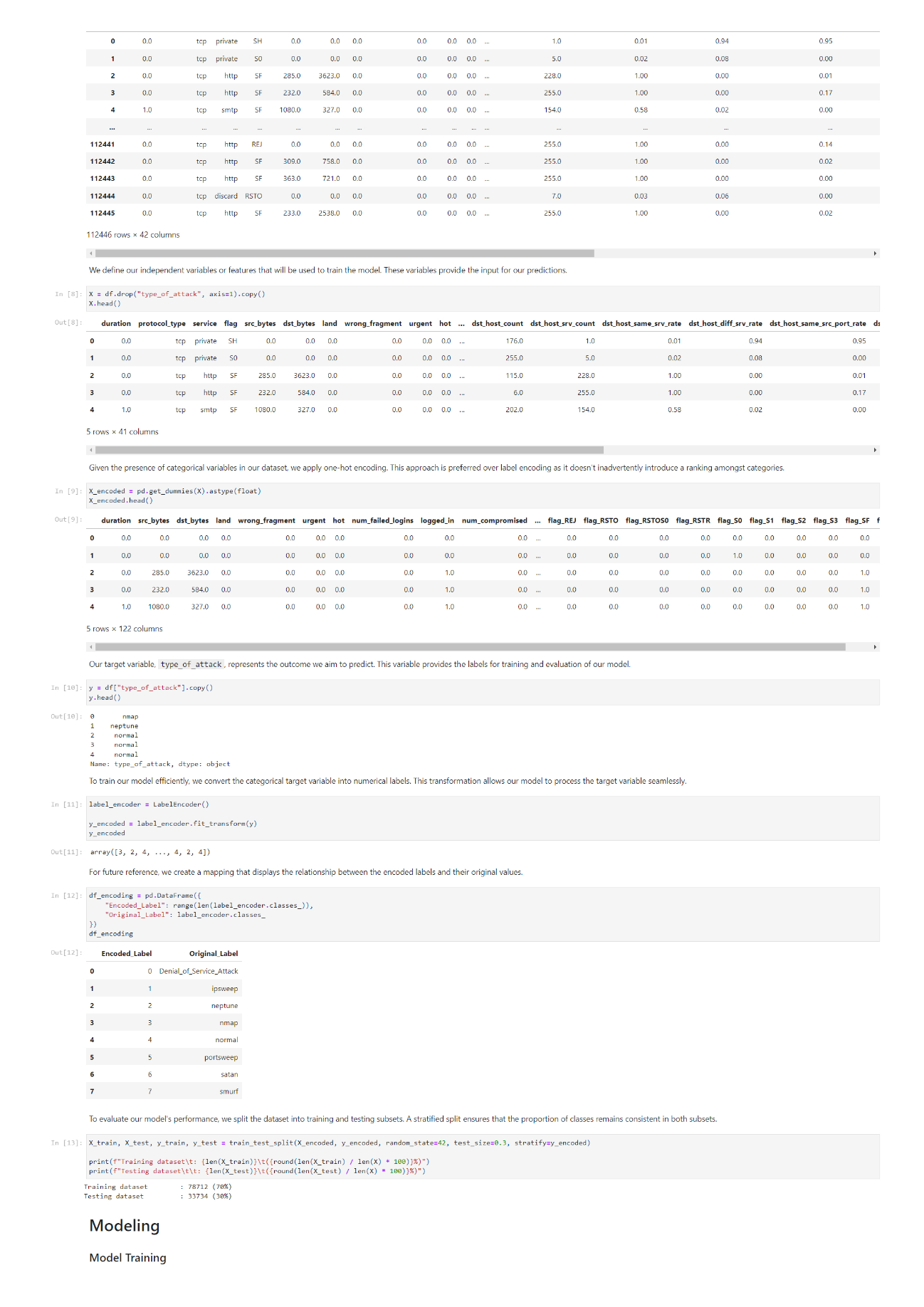
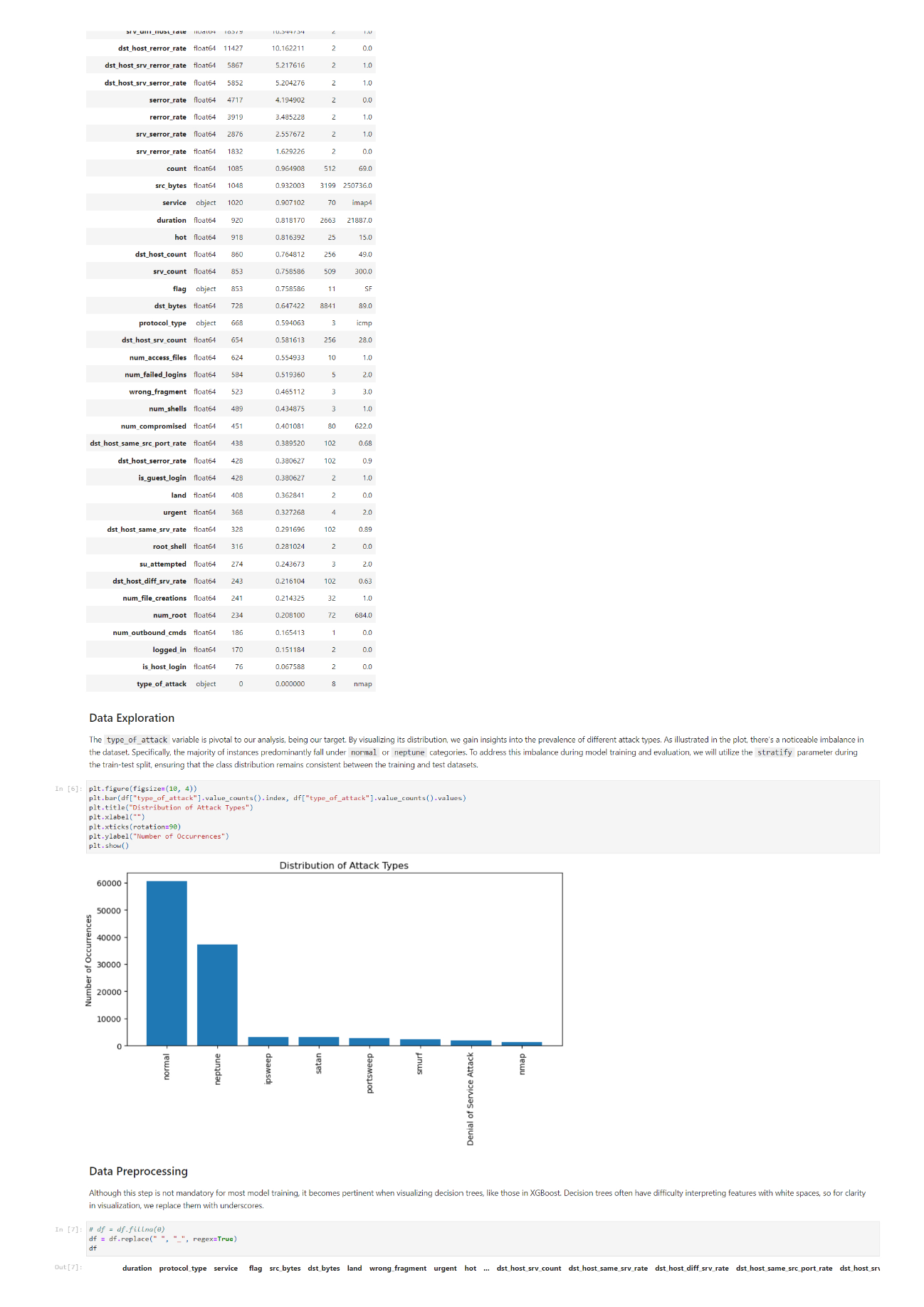


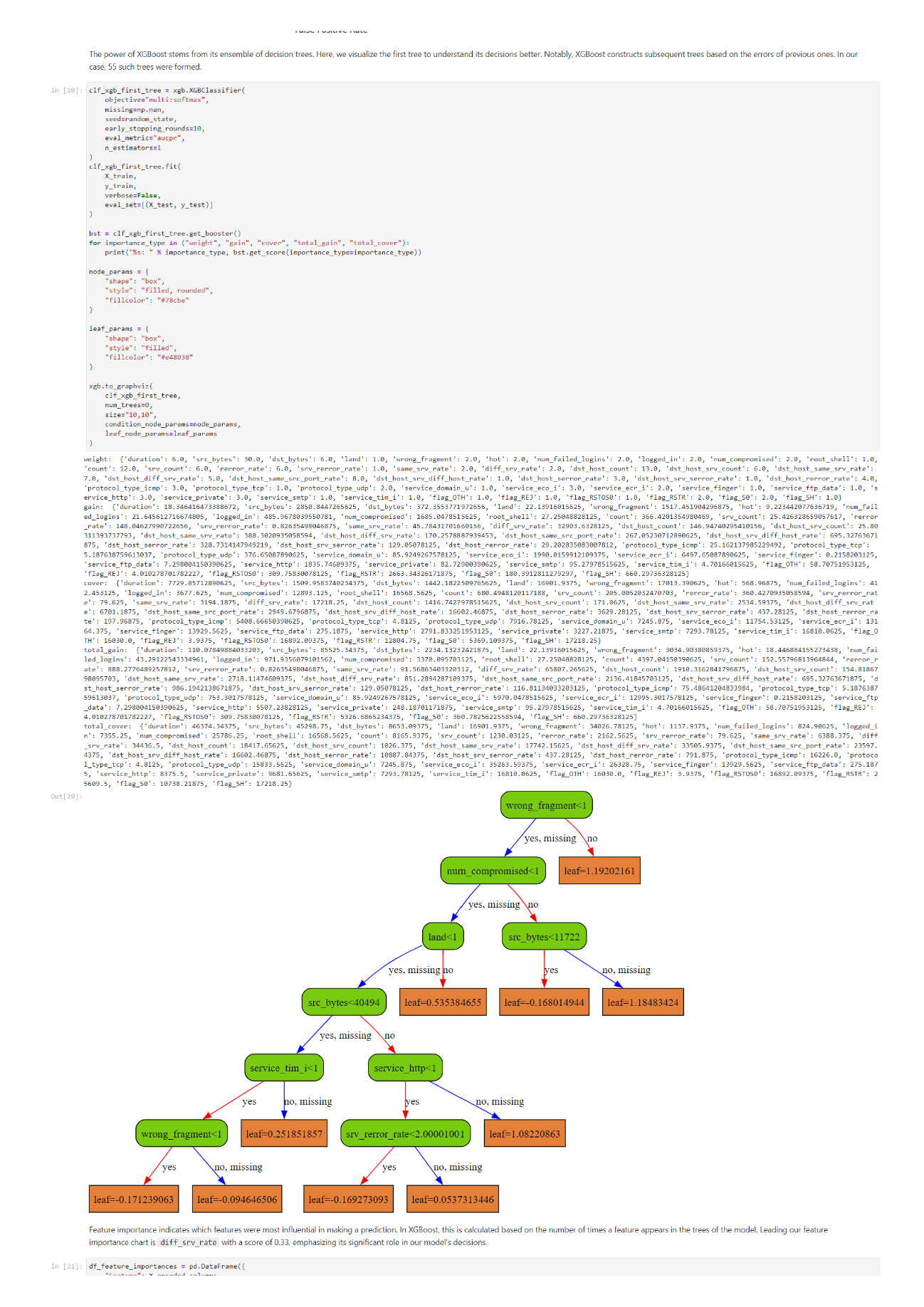
Attachment 5: First Tree in the XGBoost Model 

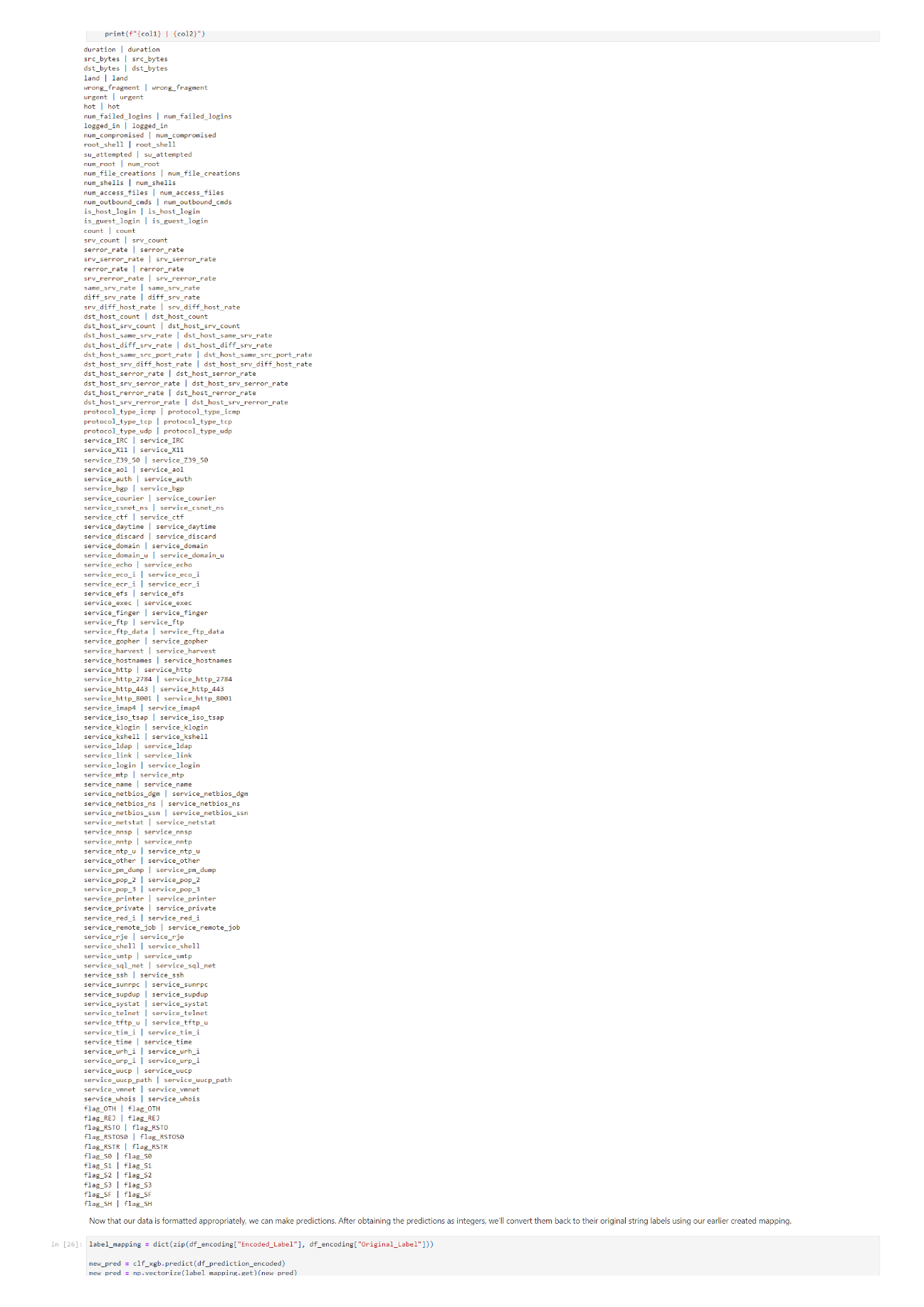
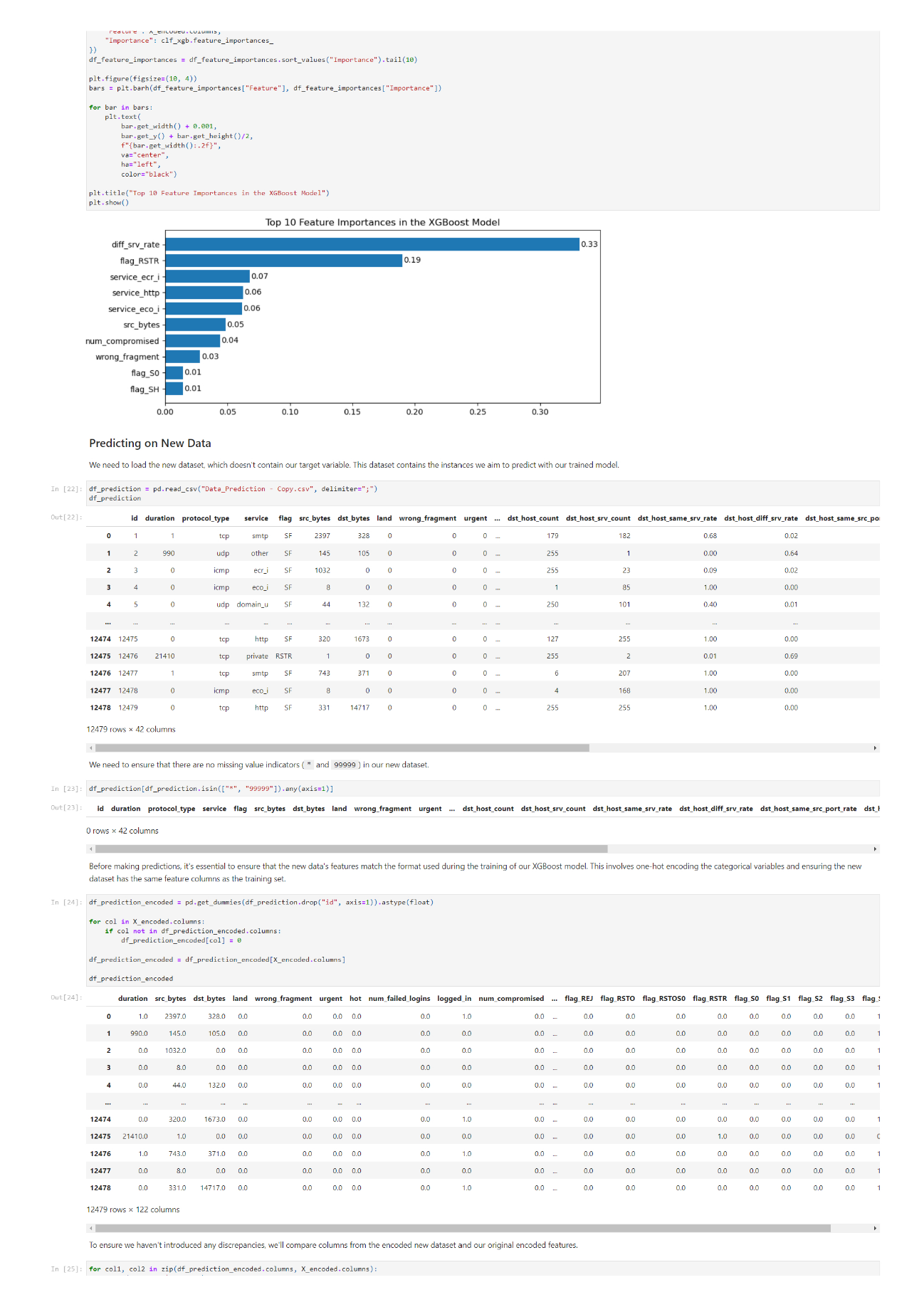
**SYNTAX**

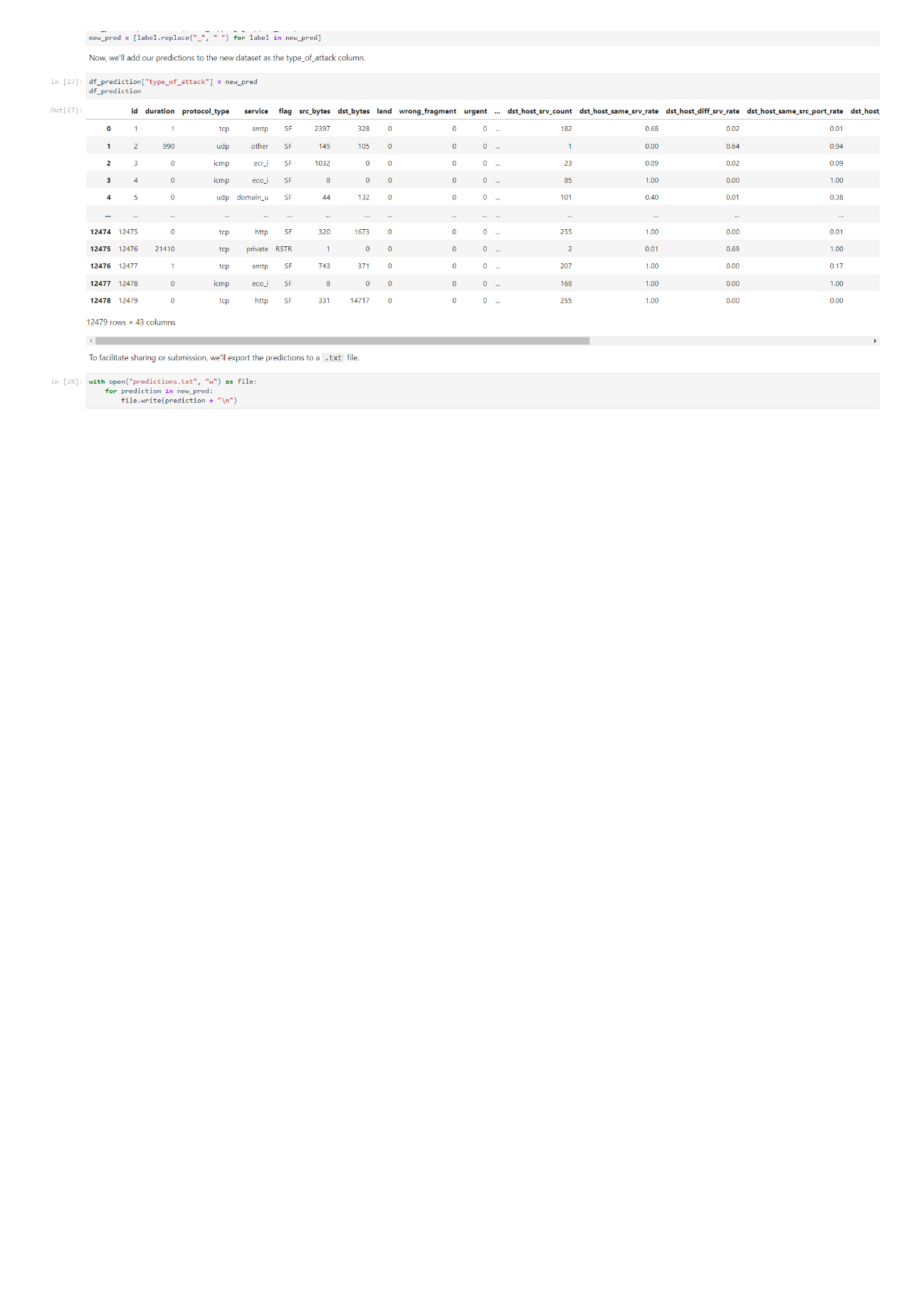
For the code used in this analysis, you can either view the Colab Notebook (Google Colab) directly [here](https://colab.research.google.com/drive/1SraFDtm27iERKvZHGH3zxKZRt12byIsh?usp=sharing) or refer to the screenshots below.

****

****

****

****

****