Intro Data Science

CSE 4095

Engineering

Malware Detection in Network Traffic

Yurim Park

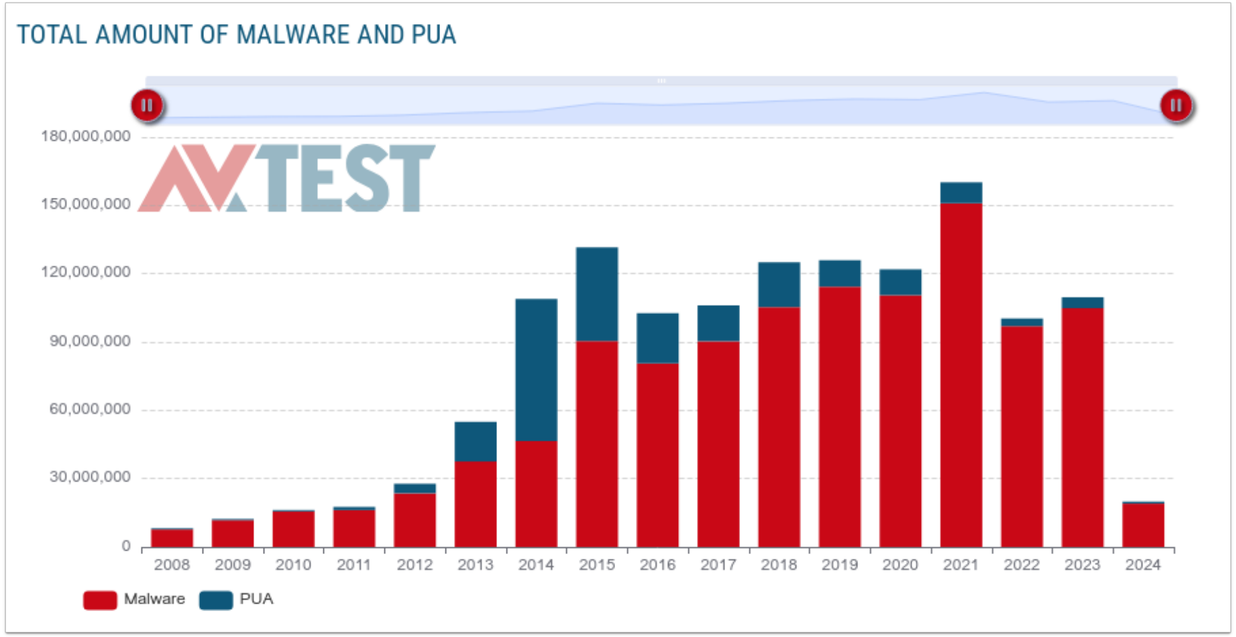
CHAPTER.1

Introduction

Background

Introduction

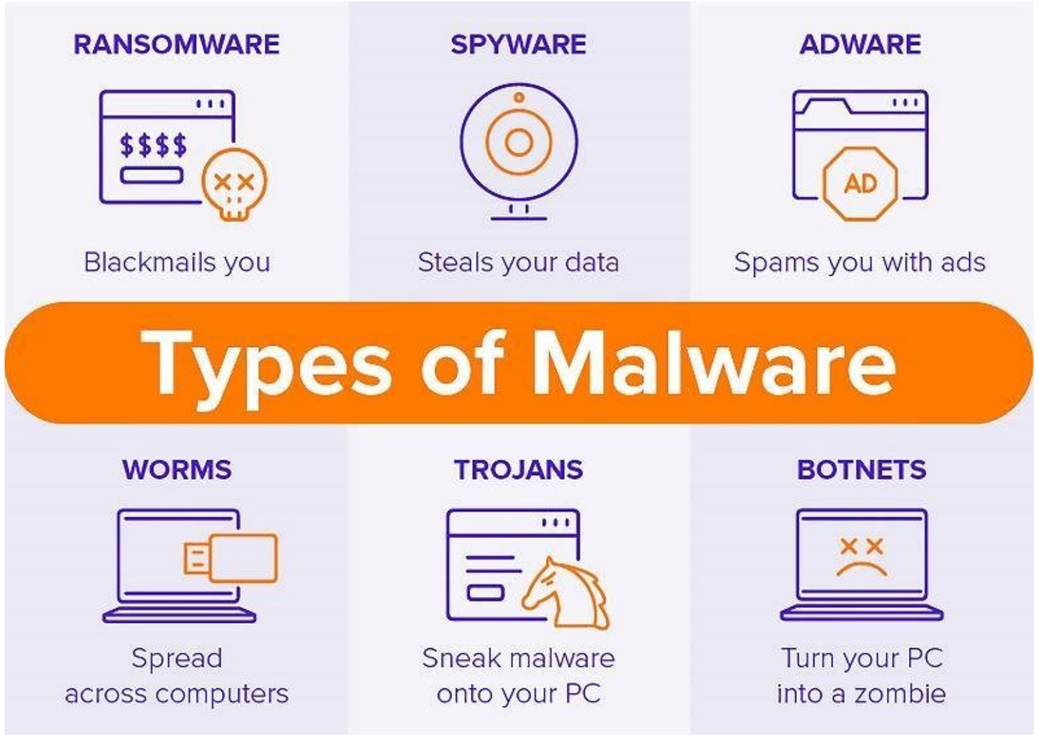
Total Amount of Malware



Malware is software designed to harm a computer system or steal sensitive information. The total amount of malware is increasing every year. Every year, more and more malware variants are created, distributed, and deployed against organizations and individuals.

Background

Types of Malware



It comes in various forms, including viruses, worms, trojan horses, spyware, and ransomware.

Motivation and Applicaiton

Detecting Malware

Given this increase in malware incidents, it's more important than ever to detect malware in network traffic. Today's systems are interconnected, and malware often spreads through these connections. By monitoring network traffic, we can identify and isolate threats before they cause significant damage.

To explore the efficacy of machine learning algorithms in identifying these malicious flows, thus enhancing network security measures

CHAPTER.2

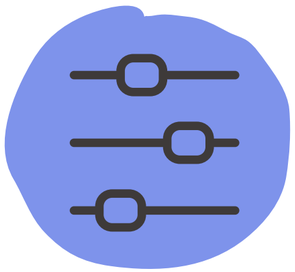
Methodology

Dataset and Collection

Dataset Labels

Attack

Benign



C&C (Command and Control)

DDoS

FileDownload

Dataset and Collection

Dataset Fields

Field	Description	Type
ts	The timestamp of the connection event.	time
uid	A unique identifier for the connection.	string
id.orig_h	The source IP address.	addr
id.orig_p	The source port.	port
id.resp_h	The destination IP address.	addr
id.resp_p	The destination port.	port
proto	The network protocol used (e.g., 'tcp').	enum
service	The service associated with the connection.	string

Dataset and Collection

Dataset Fields

Field	Description	Type
duration	The duration of the connection.	interval
orig_bytes	The number of bytes sent from the source to the destination.	count
resp_bytes	The number of bytes sent from the destination to the source.	count
conn_state	The state of the connection.	string
local_orig	Indicates whether the connection is considered local or not.	bool
local_resp	Indicates whether the connection is considered local or not.	bool
missed_bytes	The number of missed bytes in the connection.	count
label	A label associated with the connection (e.g., 'Malicious' or 'Benign').	string

Data Preprocessing

Data Preprocessing

Removing Columns

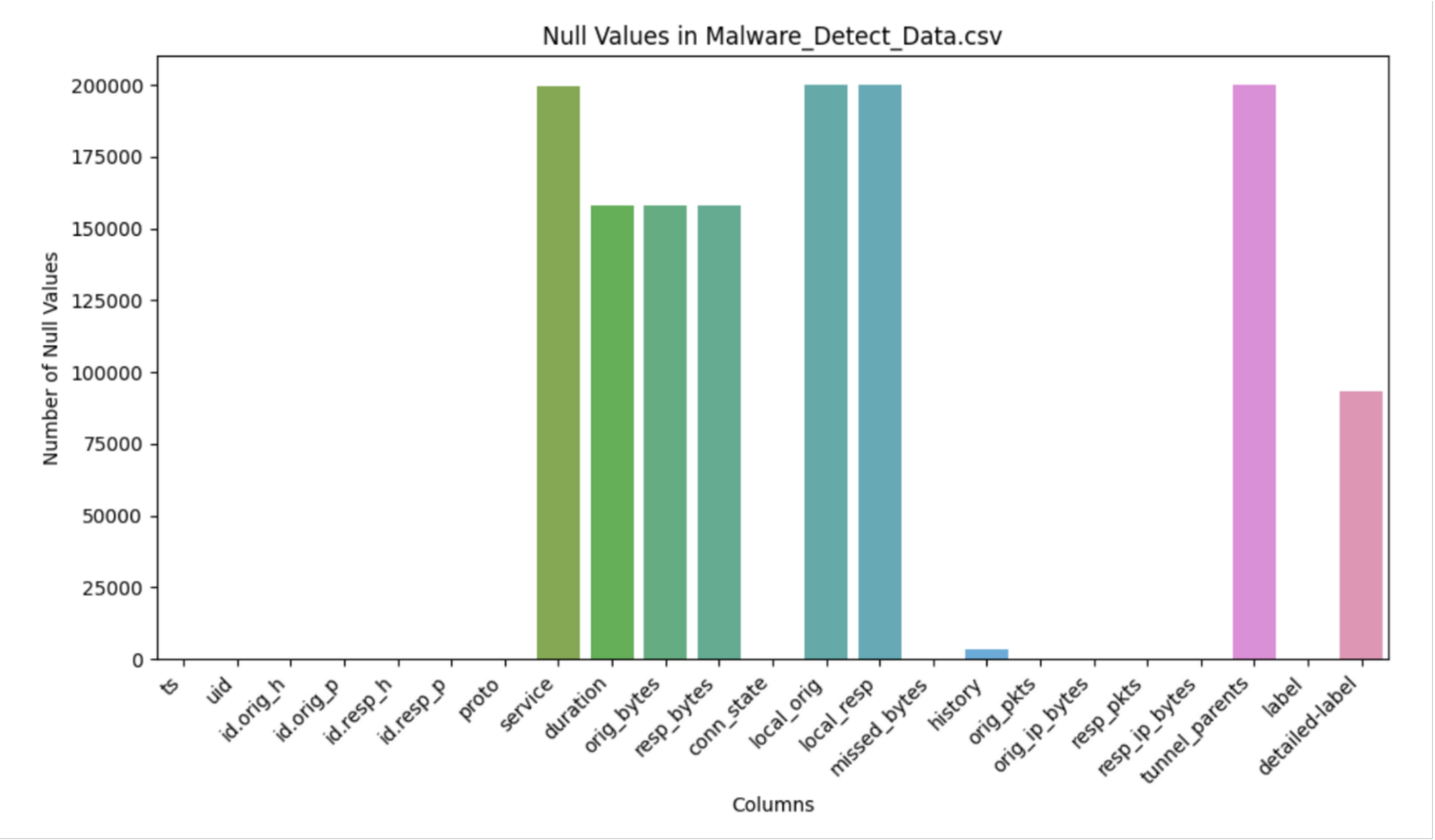
Label Encoding

Replacing Specific Values

Type Conversion

Data Preprocessing

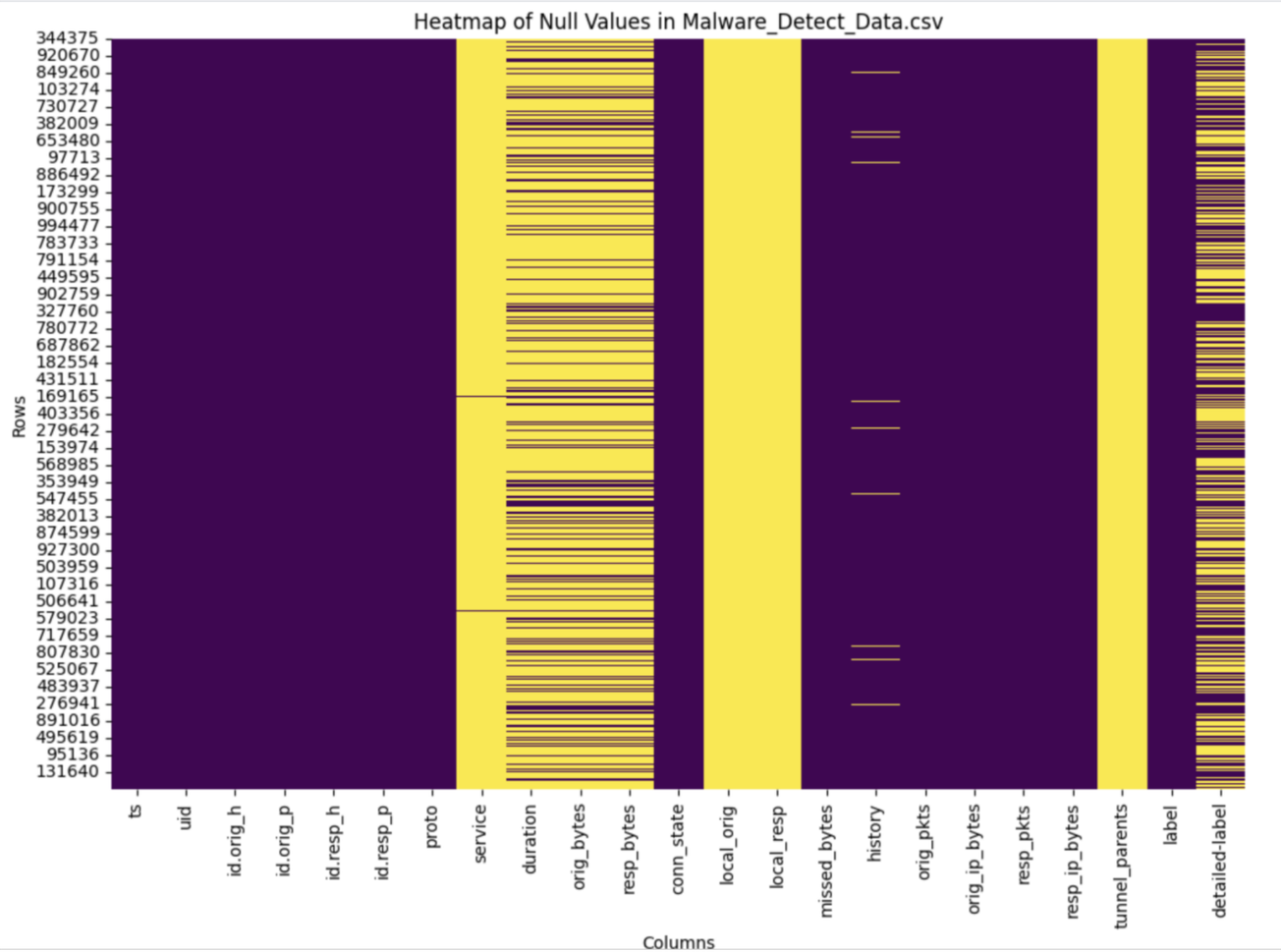
Removing Columns



Handling null values

Data Preprocessing

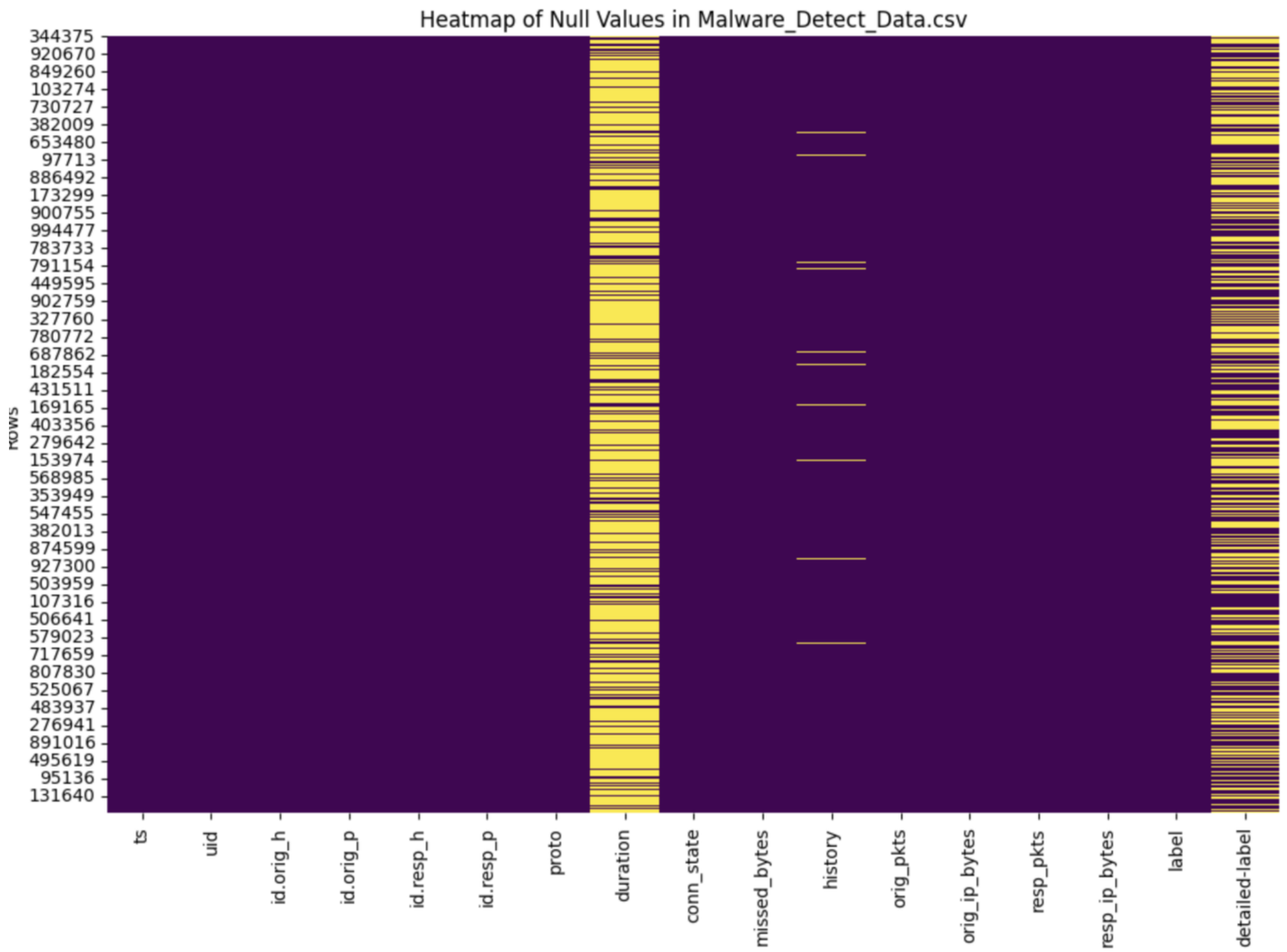
Removing Columns



Handling null values

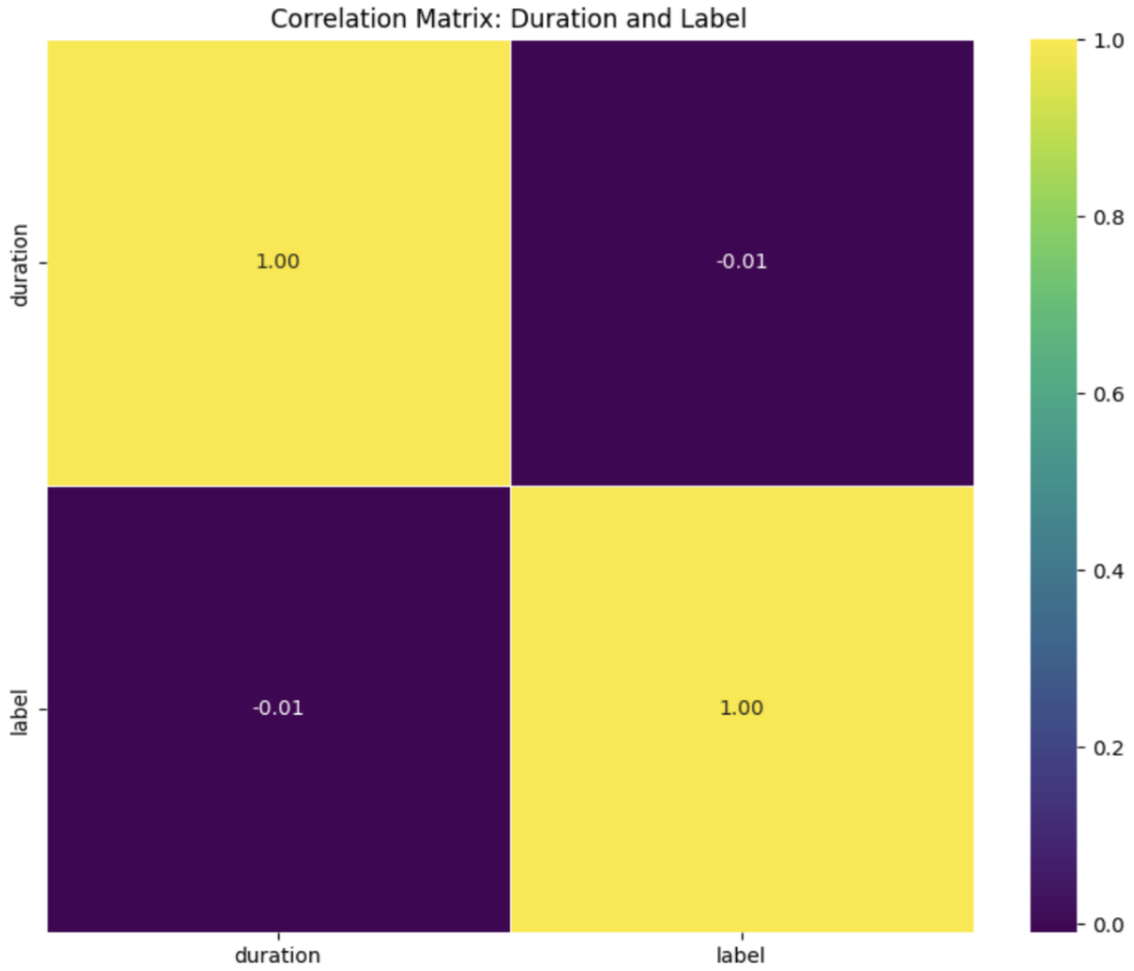
Data Preprocessing

Removing Columns



Data Preprocessing

Removing Columns



Data Preprocessing

One-Hot Encoding

Label Encoding

Food Name	Categorical #	Calories
Apple	1	95
Chicken	2	231
Broccoli	3	50

→

One Hot Encoding

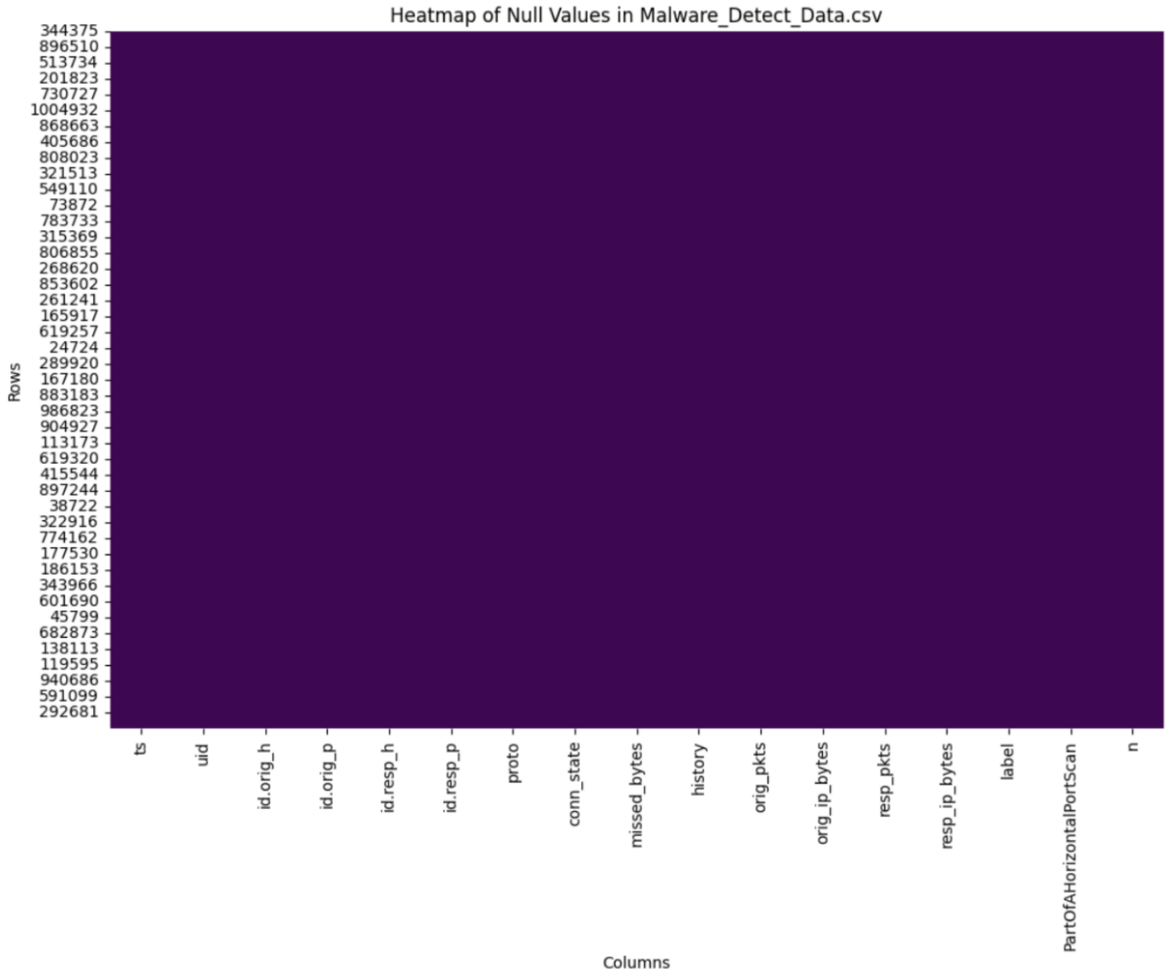
Apple	Chicken	Broccoli	Calories
1	0	0	95
0	1	0	231
0	0	1	50

Before one-hot encoding features:
'123' '123' 'udp' 'Unkown' '0.00549' '48' '48' 'SF' 'Dd' '1' '76' '1''76' '0']
(14,)
After one-hot encoding features:
'123' '123' '0.0' '0.0' '1.0' '1.0' '0.0' '0.00549' '48' '48' '0.0' '0.0'
'0.0' '0.0' '0.0' '1.0' '0.0' '0.0' '1.0' '0.0' '0.0' '0.0' '0.0' '0.0'
'0.0' '0.0' '0.0' '1' '76' '1' '76' '0']
(32,)

proto,

Data Preprocessing

Removing Columns



Data Preprocessing

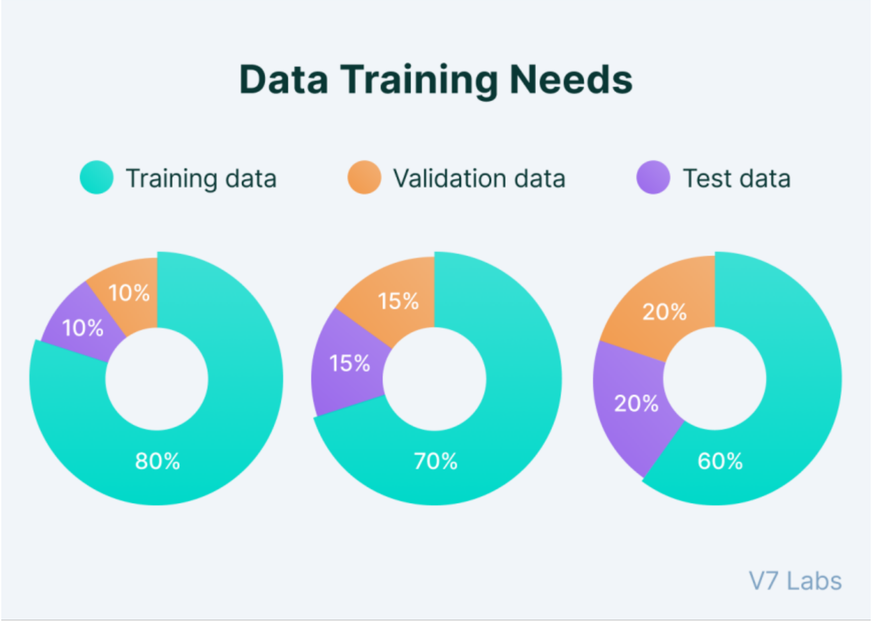
Removing Columns

```
columns_to_convert_to_float = [4]
# Convert columns to float
for row in data:
    for column in columns_to_convert_to_float:
        row[column] = float(row[column])

columns_to_convert_to_int = [0, 1, 5, 6, 8, 9, 10, 11]

# Convert columns to int
for row in data:
    for column in columns_to_convert_to_int:
        try:
            # Attempt to convert the value to an integer
            row[column] = int(row[column])
        except (ValueError, TypeError):
            pass
```

Data Preprocessing

Separating Data

splitting a dataset into training, validation, and test sets



Model Training

Classifiers

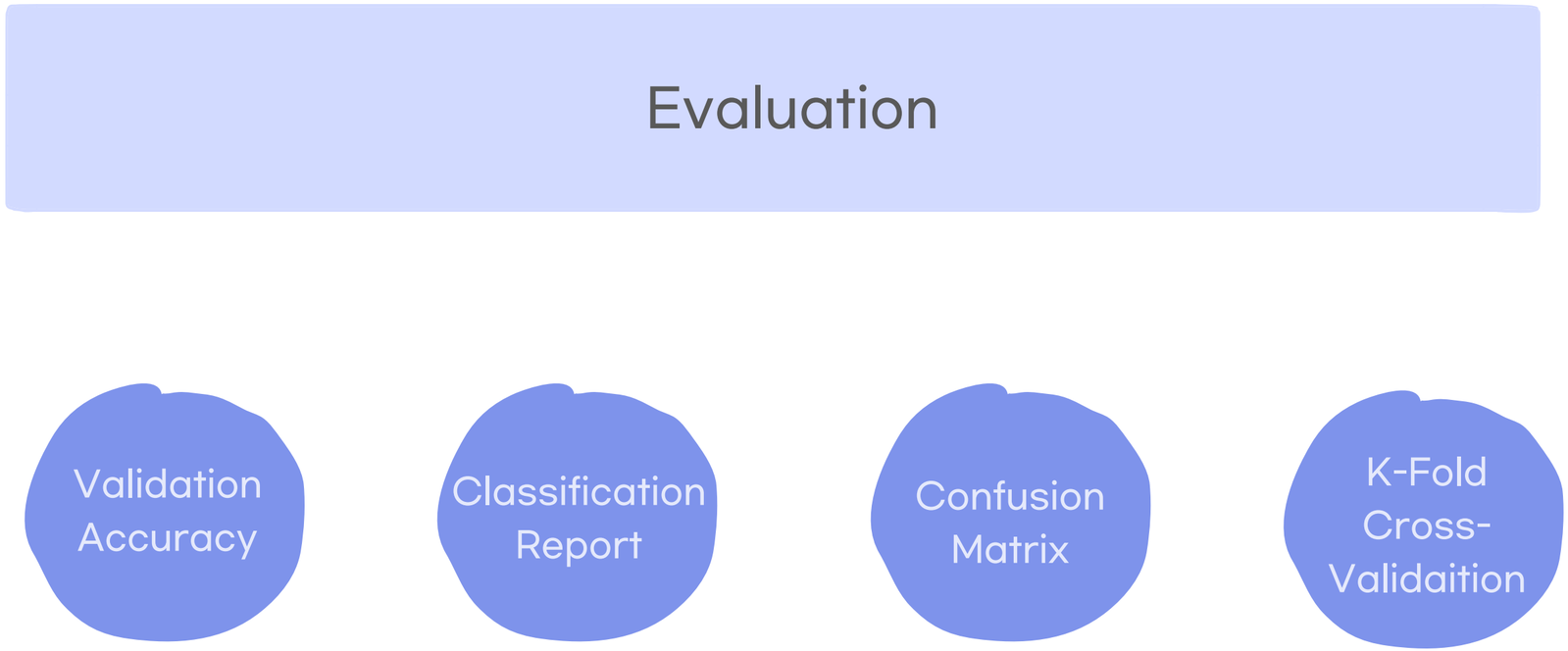
SVM (Support Vector Machine)

Support Vector Machines (SVM) are a type of supervised learning algorithm used for classification and regression tasks. The core idea of the SVM is to find a hyperplane that best separates different classes of data.

Naive Bayes

Naive Bayes is a family of probabilistic classifiers based on Bayes' Theorem. It assumes that features are conditionally independent, given the class.

Model Evaluation



Model Evaluation


```
# Evaluate the model
accuracy = accuracy_score(y_test, y_pred)
report = classification_report(y_test, y_pred)

print(f"Accuracy: {accuracy}")
print(f"Classification Report:\n{report}")
```

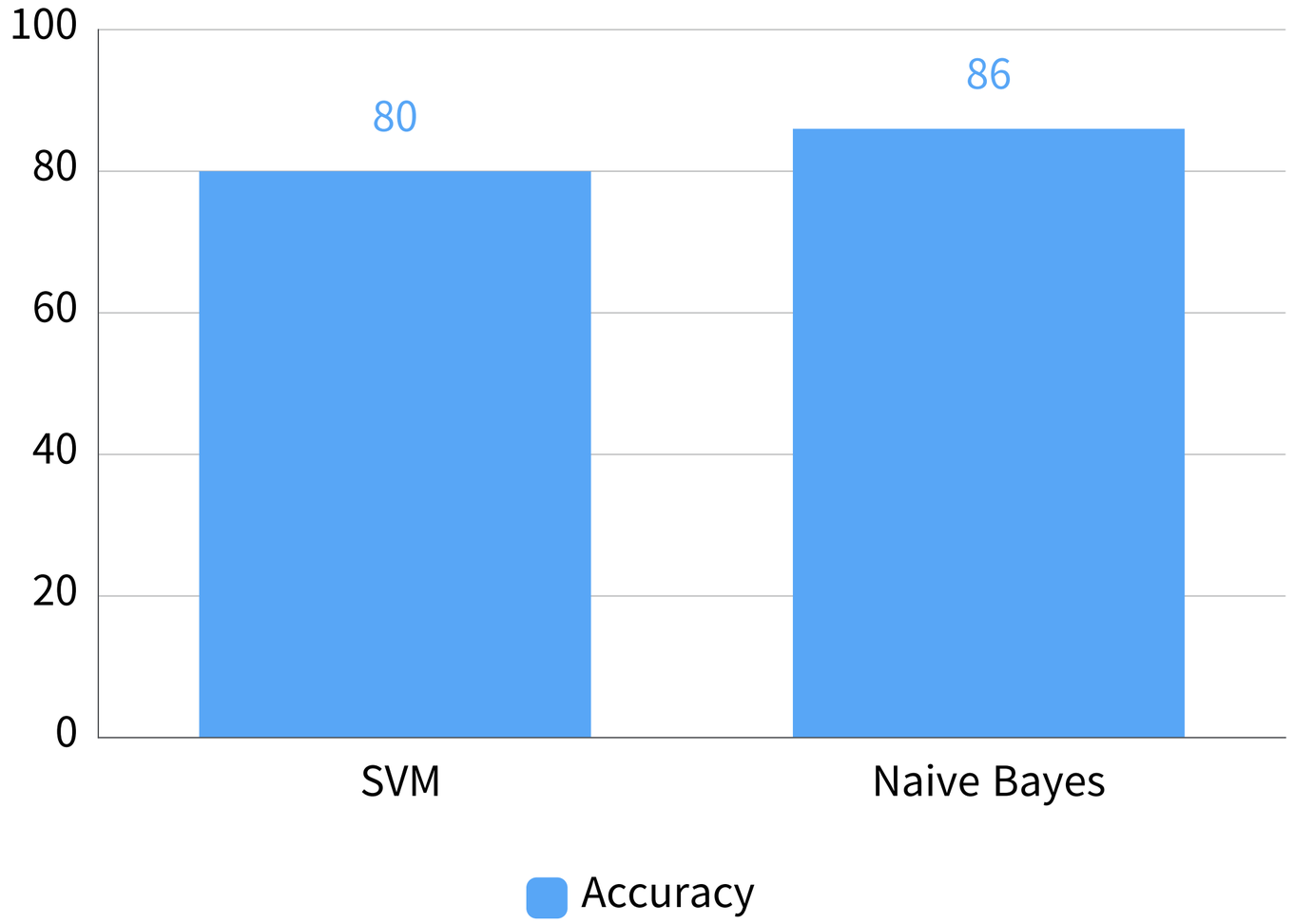
```
from sklearn.model_selection import cross_val_score, KFold


# number of folds for cross-validation
k_folds = 10

kf = KFold(n_splits=k_folds, shuffle=True, random_state=42)

clf_svm =svm_model

# performing k-fold cross-validation
cross_val_results = cross_val_score(clf_svm, X_train_scaled, y_train, c
v=kf, scoring='accuracy')

# results
print(f'Cross-validation results: {cross_val_results}')
print(f'Mean accuracy: {cross_val_results.mean()}')
```

```
from sklearn.metrics import confusion_matrix
import matplotlib.pyplot as plt

# Compute the confusion matrix
conf_matrix_nb = confusion_matrix(y_test, y_pred)

# Display the confusion matrix using a heatmap
plt.figure(figsize=(8, 6))
sns.heatmap(conf_matrix_nb, annot=True, fmt='d', cmap='Blues', xticklab
els=['0', '1'], yticklabels=['0', '1'])
plt.xlabel('Predicted Label')
plt.ylabel('True Label')
plt.title('Confusion Matrix - SVM')
plt.show()
```

CHAPTER.3

Discussion

Evaluation of Results



Evaluation of Results

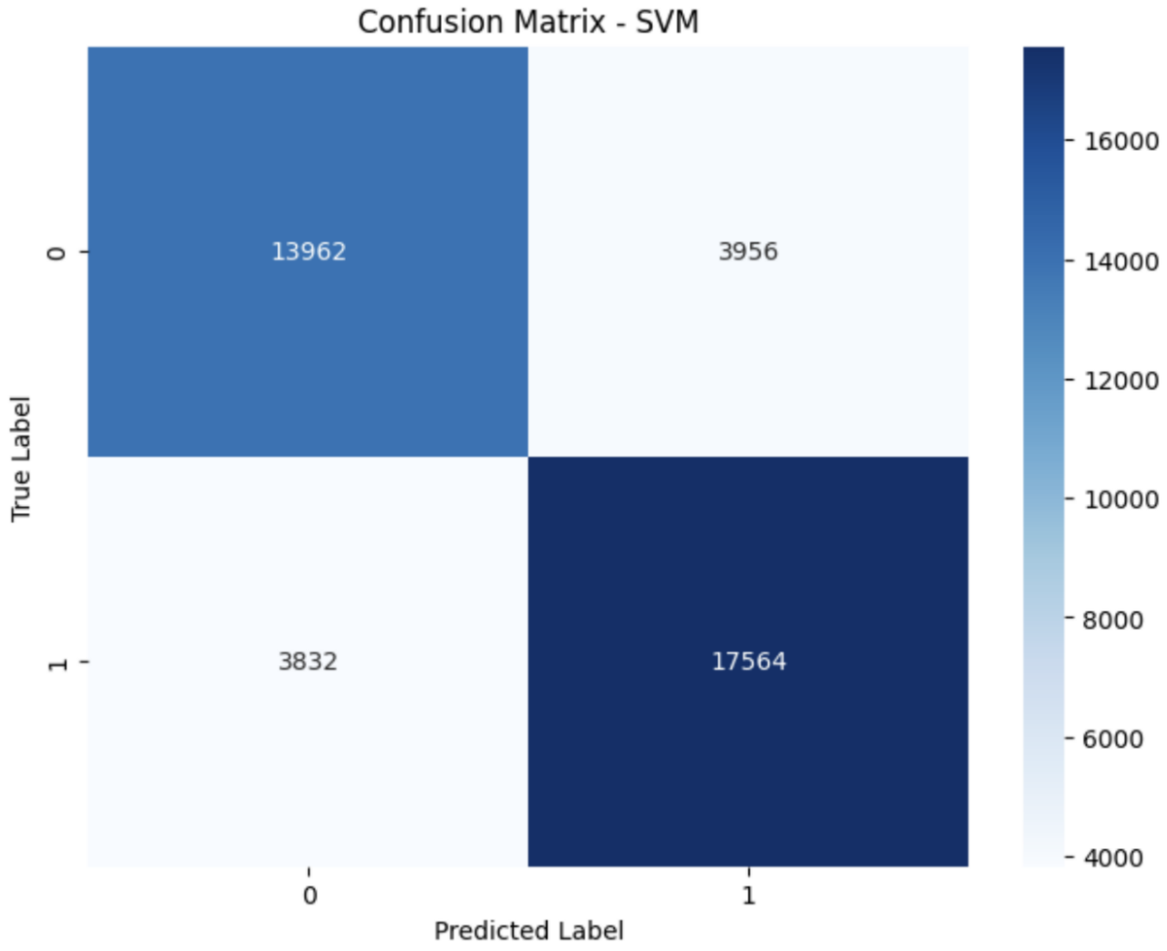
Discussion

SVM classifier

Accuracy: 0.8019026301063235
Classification Report:

	precision	recall	f1-score	support
0	0.78	0.78	0.78	17918
1	0.82	0.82	0.82	21396
accuracy			0.80	39314
macro avg	0.80	0.80	0.80	39314
weighted avg	0.80	0.80	0.80	39314

Cross-validation results: [0.54508457 0.54540252 0.53958665 0.54518283 0.54429253 0.54041335
 0.5463275 0.54759936 0.53933227 0.54689984]
Mean accuracy: 0.5440121431663502



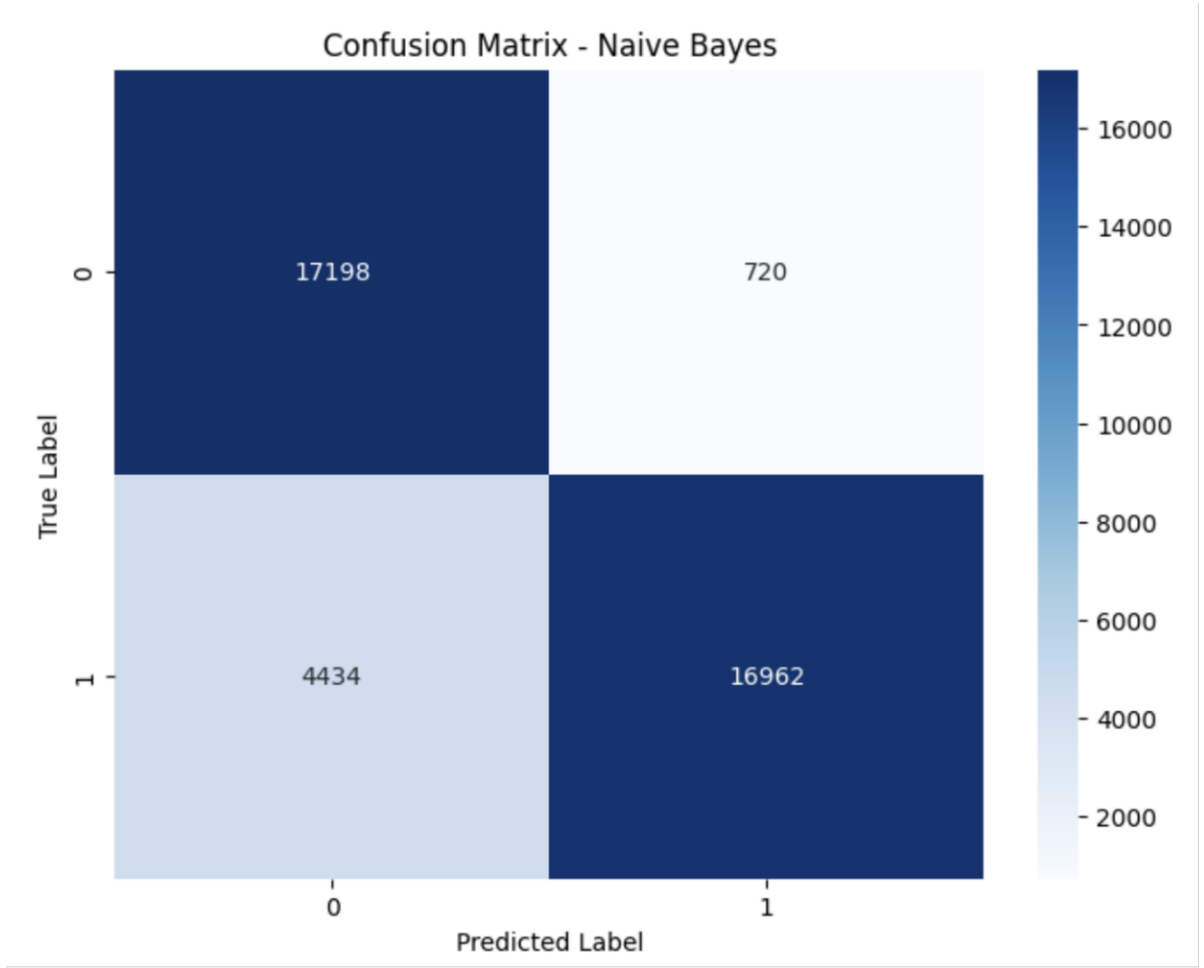
Evaluation of Results

Discussion

Naive Bayes classifier

```
Accuracy (Naive Bayes): 0.8689016635295315
Classification Report (Naive Bayes):
              precision    recall  f1-score   support

           0       0.80      0.96      0.87     17918
           1       0.96      0.79      0.87     21396

    accuracy                           0.87     39314
   macro avg       0.88      0.88      0.87     39314
weighted avg       0.88      0.87      0.87     39314
```

```
Cross-validation results: [0.93691975 0.92490144 0.93647059 0.924006
36 0.93869634 0.89825119
 0.930938   0.92559618 0.93265501 0.92400636]
Mean accuracy: 0.9272441219638294
```



Thank you for Listening