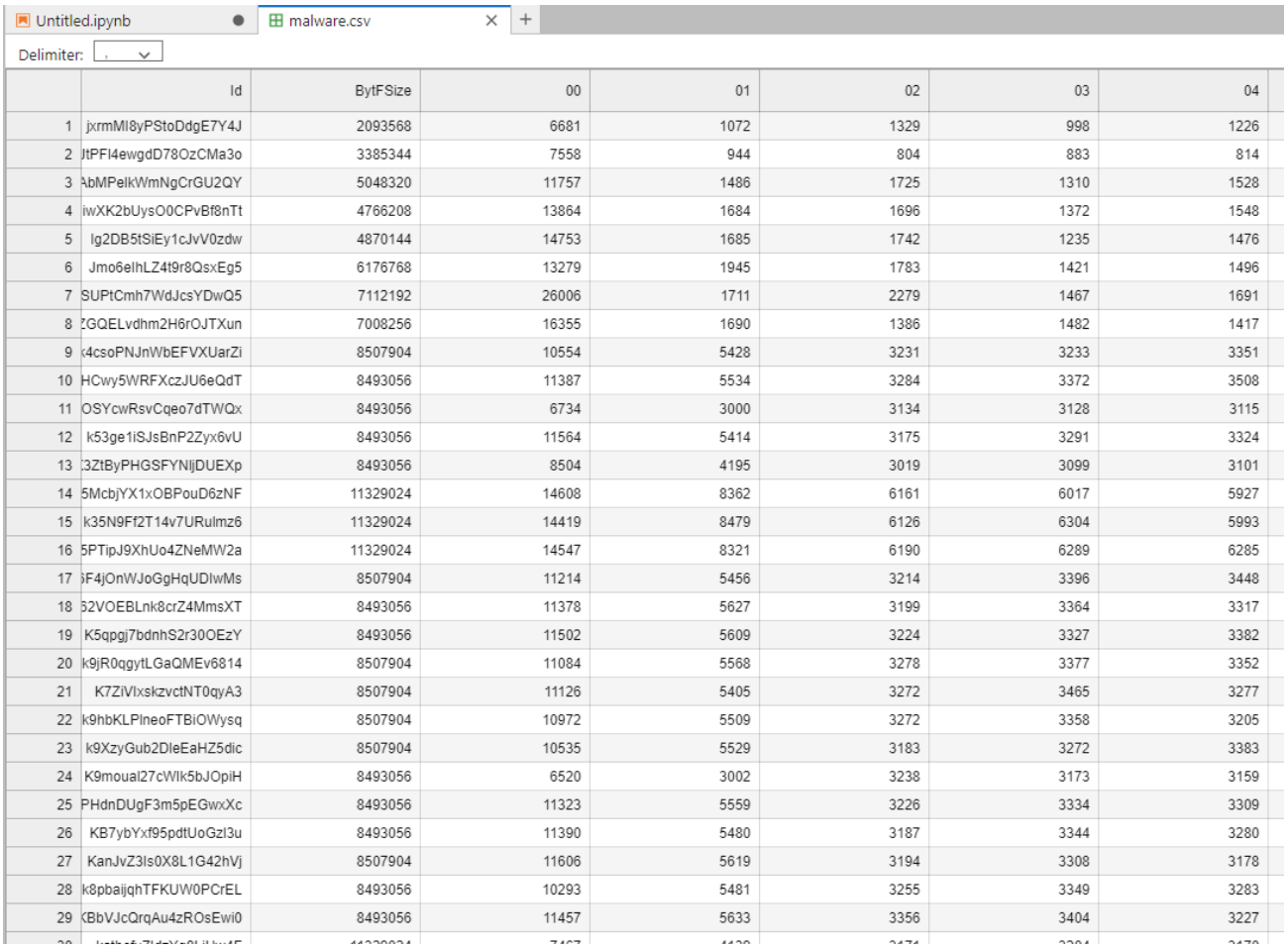
Task 1

A feed-forward neural network (FFNN), also known as a multilayer perceptron (MLP), is a foundational architecture in artificial neural networks (ANNs) and machine learning. It comprises interconnected layers of artificial neurons organized in a forward direction, where information flows from the input layer through one or more hidden layers to the output layer.

The FFNN architecture consists of three main types of layers: input layer, hidden layers, and output layer. Each layer contains multiple neurons, also referred to as nodes or units, which are interconnected through weighted connections.

FFNNs find numerous applications in various domains, including image recognition, natural language processing, finance, and cybersecurity. In cybersecurity, FFNNs are used for tasks such as intrusion detection, malware detection, phishing detection, and network traffic analysis.

FFNNs are versatile and widely-used models in machine learning, capable of learning complex patterns and making predictions or classifications based on input data.



Untitled.ipynb ● malware.csv ×

Delimiter: ,

	Id	BytFSize	00	01	02	03	04
1	jxrmMI8yPStoDdgE7Y4J	2093568	6681	1072	1329	998	1226
2	JtPFI4ewgdD78OzCMa3o	3385344	7558	944	804	883	814
3	AbMPelkWmNgCrGU2QY	5048320	11757	1486	1725	1310	1528
4	iwXK2bUysO0CPvBf8nTt	4766208	13864	1684	1696	1372	1548
5	Ig2DB5tSiEy1cJvV0zdw	4870144	14753	1685	1742	1235	1476
6	Jmo6elhLZ4t9r8QsxEg5	6176768	13279	1945	1783	1421	1496
7	SUPtCmh7WdJcsYDwQ5	7112192	26006	1711	2279	1467	1691
8	ZGQELvdhm2H6rOJTXun	7008256	16355	1690	1386	1482	1417
9	x4csoPNJnWbEFVXUarZi	8507904	10554	5428	3231	3233	3351
10	HCwy5WRFXczJU6eQdT	8493056	11387	5534	3284	3372	3508
11	OSYcwRsvCqeo7dTWQx	8493056	6734	3000	3134	3128	3115
12	k53ge1iSJsBnP2Zyx6vU	8493056	11564	5414	3175	3291	3324
13	3ZtByPHGSFYNIjDUEXp	8493056	8504	4195	3019	3099	3101
14	5McbjYX1xOBPouD6zNF	11329024	14608	8362	6161	6017	5927
15	k35N9Ff2T14v7URuImz6	11329024	14419	8479	6126	6304	5993
16	5PTipJ9XhUo4ZNeMW2a	11329024	14547	8321	6190	6289	6285
17	5F4jOnWJoGgHqUDIwMs	8507904	11214	5456	3214	3396	3448
18	52VOEBLnk8crZ4MmsXT	8493056	11378	5627	3199	3364	3317
19	K5qpgj7bdnhS2r30OEzY	8493056	11502	5609	3224	3327	3382
20	k9jR0qgytLGaQMEv6814	8507904	11084	5568	3278	3377	3352
21	K7ZiVIxskzvctNT0qyA3	8507904	11126	5405	3272	3465	3277
22	k9hbKLPIneoFTBiOWysq	8507904	10972	5509	3272	3358	3205
23	k9XzyGub2DleEaHZ5dic	8507904	10535	5529	3183	3272	3383
24	K9moual27cWlk5bJOpiH	8493056	6520	3002	3238	3173	3159
25	PHdnDUgF3m5pEGwxXc	8493056	11323	5559	3226	3334	3309
26	KB7ybYxf95pdtUoGzI3u	8493056	11390	5480	3187	3344	3280
27	KanJvZ3Is0X8L1G42hVj	8507904	11606	5619	3194	3308	3178
28	k8pbaijqhTFKUW0PCrEL	8493056	10293	5481	3255	3349	3283
29	KBbVJcQrqAu4zROsEwi0	8493056	11457	5633	3356	3404	3227

Importing required libraries

```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.neural_network import MLPClassifier
from sklearn.metrics import accuracy_score

# Load the dataset
data = pd.read_csv("malware.csv")

# Preprocessing
X = data.drop(columns=['legitimate'])
y = data['legitimate']

# Splitting the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Feature scaling
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

# Creating and training the feed-forward neural network model
mlp = MLPClassifier(hidden_layer_sizes=(100, 50), activation='relu', solver='adam', random_state=42)
mlp.fit(X_train_scaled, y_train)

# Predictions
y_pred = mlp.predict(X_test_scaled)
```

```
# Evaluation
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)
```