BIRZEIT UNIVERSITY
Faculty of Engineering & Technology
Electrical & Computer Engineering Department

ARTIFICIAL INTELLIGENCE | ENCS3340

Project #2 : Machine Learning for Classification

Team Members:

Hamed Musleh	**1221036**	**Section: 1**
Yahya Sarhan	**1221858**	**Section: 1**

Instructor: Yazan Abu Farha

Submission Date: June, 18, 2025

Abstract:

This project explores the performance of three machine learning algorithms Naive Bayes, Decision Tree, and Feedforward Neural Network (FNN) in the task of image classification. A dataset consisting of categorized images of cats, dogs, and snakes was used. The images were preprocessed by resizing and flattening into 1D vectors to be compatible with the models. Each classifier was trained and evaluated using standard performance metrics such as accuracy, precision, recall, and F1-score. The goal was to analyze the strengths and limitations of each approach and determine which model performs best in terms of generalization and efficiency for this classification task.

Table of Contents

Table of Figure:

List of Table:

Algorithm Definitions

Naive Bayes Classifier:

Naive Bayes is a probabilistic classification algorithm based on Bayes' Theorem. It assumes that features are conditionally independent given the class label an assumption that is rarely true in practice but often works surprisingly well. In image classification, it uses pixel values as features and estimates the probability of each class for a given image. It is fast, requires less data for training, and serves as a solid baseline model.

Decision Tree Classifier:

A Decision Tree classifier splits data into subsets based on the value of input features, forming a tree-like structure where each node represents a decision rule and each leaf represents a class label. It is easy to interpret and visualize, making it useful for understanding feature importance. However, it can easily overfit the training data, especially when the tree grows too deep.

Feedforward Neural Network (FNN):

A Feedforward Neural Network (FNN) is a type of artificial neural network in which data flows in a single direction — from the input layer, through one or more **hidden layers**, to the **output layer**, without any cycles or loops.

Each neuron in the network performs a weighted sum of its inputs followed by an activation function, enabling the network to model nonlinear relationships. The model is trained using backpropagation along with an optimization algorithm such as Stochastic Gradient Descent (SGD) or Adam to minimize prediction errors.

FNNs are capable of capturing complex patterns and often outperform simpler models like Naive Bayes and Decision Trees in image classification tasks.

In this project, and in accordance with the project specification, the implemented neural network was configured to use **1–2 hidden layers**. This architectural choice maintains a balance between model complexity and training efficiency, while still allowing the network to learn meaningful representations of the image data.

Dataset Description

The dataset used in this project consists of **2,390 labeled images** categorized into **three** transportation-related classes: **Car, Bikes, and Plane**. Each class contains a different number of images, as shown in the table below [1] :

Table 1 : Photo Class

Class	**Number of Images**
Car	790
Bikes	800
plane	800
Total	2390

All images were **resized to a fixed resolution (e.g., 64×64 pixels)** and flattened into 1D feature vectors in order to be compatible with the input requirements of the machine learning models.

This dataset presents a diverse set of transportation types, which challenges the classifiers to generalize across visual characteristics such as **shape, size, and structure**. The slight class imbalance also enables evaluation of how different models handle uneven class distributions during training and prediction.

Result & Discussion:

1 - Naive Bayes Classifier:

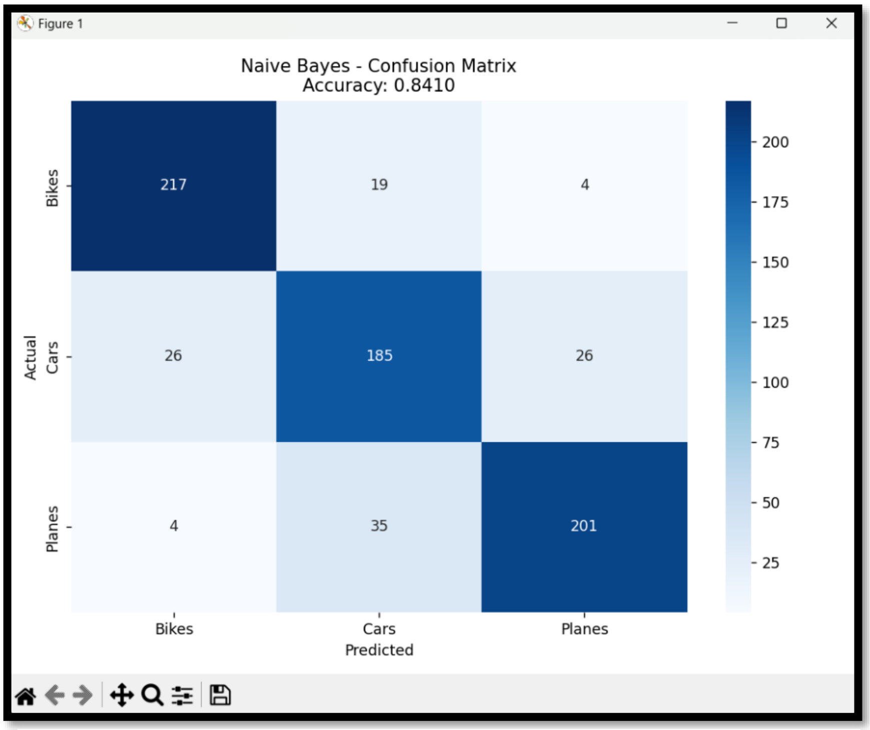

Figure 1 : Confusion Matrix for Naive Bayes classifier on the Vehicles dataset (Bikes, Cars, Planes). The overall accuracy achieved was 84.10%.

Table 2 : Naive Bayes Table

#########	Predicted: Bikes	Predicted: Cars	Predicted: Planes	Total (per class)
Actual: Bikes	217 (TP)	19 (FN)	4 (FN)	240
Actual: Cars	26 (FP)	185 (TP)	26 (FN)	237
Actual: Planes	4 (FP)	35 (FN)	201 (TP)	240

Bikes in Naive Bayes Classifier :

- **True Positive (TP):** 217
- **False Positive (FP):** 26 (Actual Cars) + 4 (Actual Planes) = 30
- **False Negative (FN):** 19 + 4 = 23

Precision = TP / (TP + FP) = 217 / (217 + 30) ≈ 0.8786
Recall = TP / (TP + FN) = 217 / (217 + 23) ≈ 0.9042
F1 Score = 2 * (P * R) / (P + R) ≈ 0.8912

Cars in Naive Bayes Classifier:

- **TP:** 185
- **FP:** 19 (Bikes) + 35 (Planes) = 54
- **FN:** 26 + 26 = 52

Precision = 185 / (185 + 54) ≈ 0.7745
Recall = 185 / (185 + 52) ≈ 0.7806
F1 Score ≈ 0.7775

Planes in Naive Bayes Classifier:

- **TP:** 201
- **FP:** 4 (Bikes) + 26 (Cars) = 30
- **FN:** 4 + 35 = 39

Precision = 201 / (201 + 30) ≈ 0.8701
Recall = 201 / (201 + 39) ≈ 0.8375
F1 Score ≈ 0.8535

Accuracy in Naive Bayes Classifier:

Accuracy = (TP total) / (Total samples) = (217 + 185 + 201) / 717 ≈ 0.8410.

The figure shows the confusion matrix for a Naive Bayes classifier trained on the Vehicles dataset, consisting of three classes: Bikes, Cars, and Planes. The model achieved an overall accuracy of **84.10%**.

- For **Bikes**, the precision was **87.86%**, recall **90.42%**, and F1-score **89.12%**.
- For **Cars**, the precision was **77.45%**, recall **78.06%**, and F1-score **77.75%**.
- For **Planes**, the precision was **87.01%**, recall **83.75%**, and F1-score **85.35%**.

The classifier performed best on the Bikes class, while slightly struggling to distinguish between Cars and Planes, as seen by the relatively high misclassifications between these two.

2- Decision Tree Classifier :

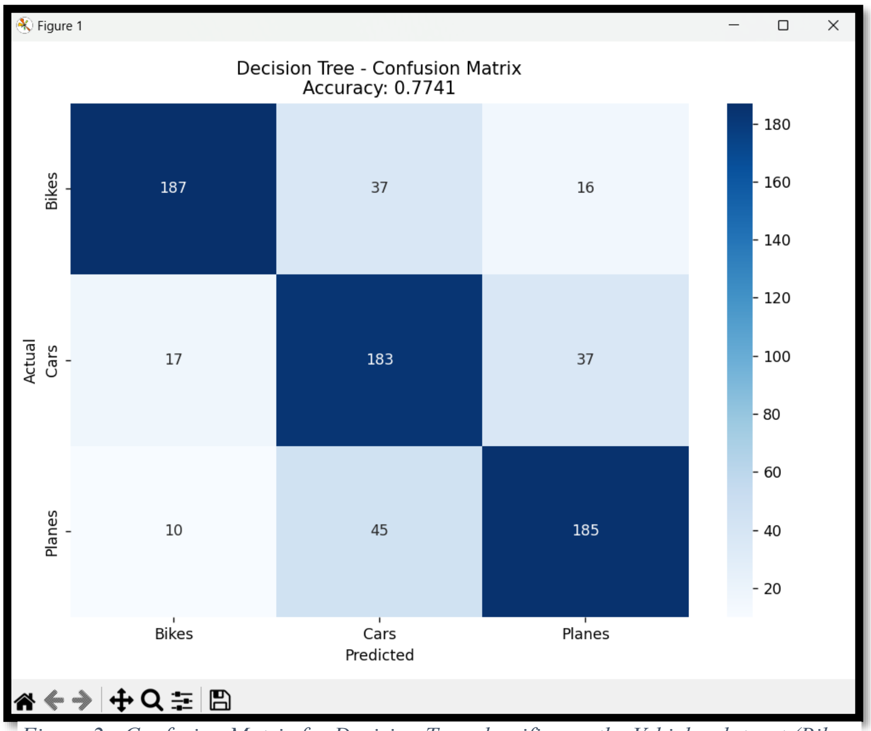

Figure 2 : Confusion Matrix for Decision Tree classifier on the Vehicles dataset (Bikes, Cars, Planes). The model achieved an overall accuracy of 77.41%.

Table 3 : Decision Tree Table

######	Predicted: Bikes	Predicted: Cars	Predicted: Planes	Total
Actual: Bikes	187	37	16	240
Actual: Cars	17	183	37	237
Actual: Planes	10	45	185	240

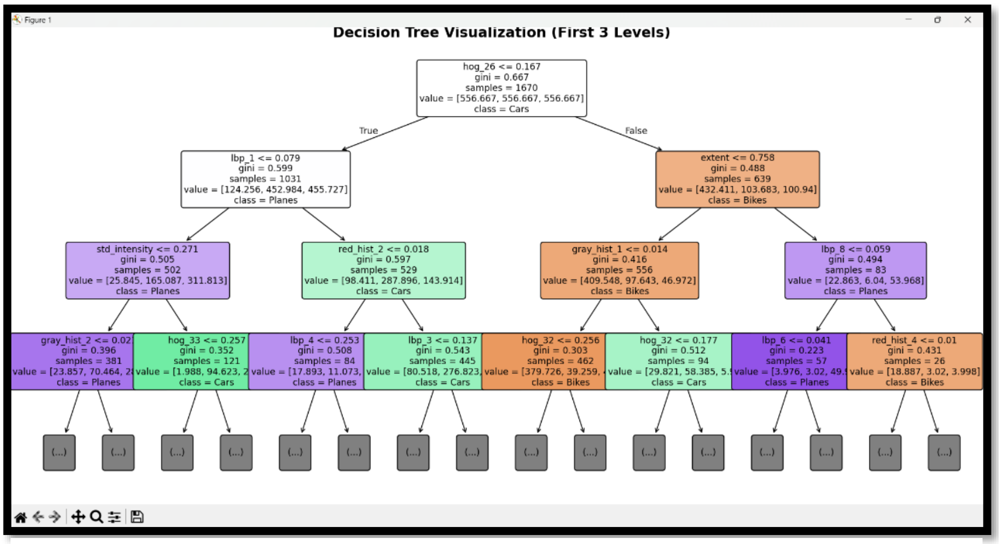

Figure 3 : Visualize Decision Tree

Bikes in Decision Tree:

- TP = 187
- FP = 17 (from Cars) + 10 (from Planes) = 27
- FN = 37 + 16 = 53

Precision = 187 / (187 + 27) ≈ 0.8739
Recall = 187 / (187 + 53) ≈ 0.7792
F1-score ≈ 0.8237.

Cars in Decision Tree:

- TP = 183
- FP = 37 (from Bikes) + 45 (from Planes) = 82
- FN = 17 + 37 = 54

Precision = 183 / (183 + 82) ≈ 0.6905
Recall = 183 / (183 + 54) ≈ 0.7721
F1-score ≈ 0.7291.

Planes in Decision Tree Classifier:

- TP = 185
- FP = 16 (from Bikes) + 37 (from Cars) = 53
- FN = 10 + 45 = 55

Precision = 185 / (185 + 53) ≈ 0.7774
Recall = 185 / (185 + 55) ≈ 0.7708
F1-score ≈ 0.7741.

Accuracy in Decision Tree Classifier :

Accuracy = (TP total) / (Total samples) = (187 + 183 + 185) / 717 ≈ 0.7741.

The figure illustrates the confusion matrix for a Decision Tree classifier applied to the Vehicles dataset, consisting of three categories: Bikes, Cars, and Planes.

The model achieved an overall accuracy of **77.41%**, showing moderate performance across the classes.

- For **Bikes**, the model achieved a precision of **87.39%**, recall of **77.92%**, and F1-score of **82.37%**.
- For **Cars**, the precision was **69.05%**, recall **77.21%**, and F1-score **72.91%**.
- For **Planes**, the model had a precision of **77.74%**, recall **77.08%**, and F1-score **77.41%**.

Compared to Naive Bayes, the Decision Tree classifier was less accurate overall, especially in classifying **Cars**, where it had a noticeably lower precision due to misclassification with Planes.

3 - Feedforward Neural Network :

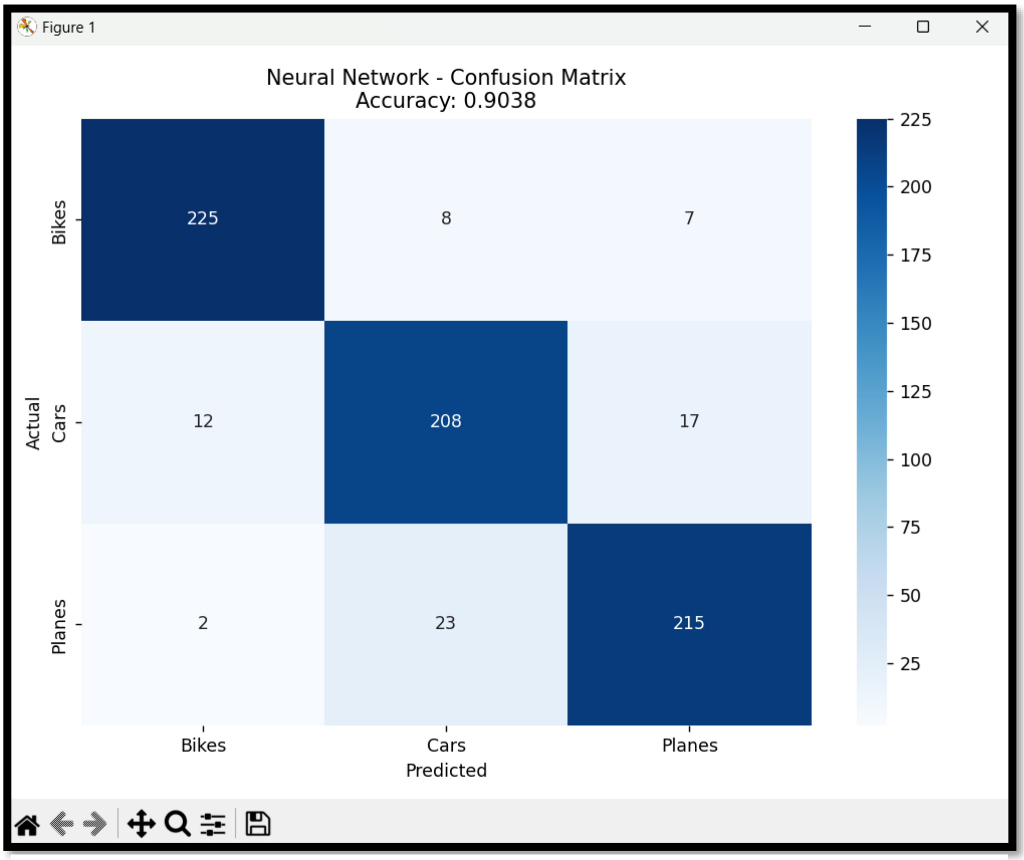

Figure 4 : Confusion Matrix for Neural Network classifier on the Vehicles dataset (Bikes, Cars, Planes). The model achieved an overall accuracy of 90.38%.

Table 4 : Feedforward Neural Network Table

########	**Predicted: Bikes**	**Predicted: Cars**	**Predicted: Planes**	**Total**
Actual: Bikes	225	8	7	240
Actual: Cars	12	208	17	237
Actual: Planes	2	23	215	240

Bikes in Feedforward Neural Network

- TP = 225
- FP = 12 (from Cars) + 2 (from Planes) = 14
- FN = 8 + 7 = 15

Precision = 225 / (225 + 14) ≈ 0.9414
Recall = 225 / (225 + 15) ≈ 0.9375
F1-score ≈ 0.9395

Cars in Feedforward Neural Network:

- TP = 208
- FP = 8 (from Bikes) + 23 (from Planes) = 31
- FN = 12 + 17 = 29

Precision = 208 / (208 + 31) ≈ 0.8702
Recall = 208 / (208 + 29) ≈ 0.8772
F1-score ≈ 0.8737

Plane in Feedforward Neural Network:

- TP = 215
- FP = 7 (from Bikes) + 17 (from Cars) = 24
- FN = 2 + 23 = 25

Precision = 215 / (215 + 24) ≈ 0.8995
Recall = 215 / (215 + 25) ≈ 0.8958
F1-score ≈ 0.8976.

The confusion matrix above shows the classification performance of a Neural Network model on the Vehicles dataset, which includes three categories: Bikes, Cars, and Planes. The model achieved the highest performance among the tested classifiers, with an overall accuracy of **90.38%**.

- For **Bikes**, the model reached a precision of **94.14%**, recall **93.75%**, and F1-score **93.95%**.
- For **Cars**, the precision was **87.02%**, recall **87.72%**, and F1-score **87.37%**.
- For **Planes**, the model achieved a precision of **89.95%**, recall **89.58%**, and F1-score **89.76%**.

The Neural Network demonstrated strong and balanced performance across all three classes, with minimal confusion between categories compared to other models.

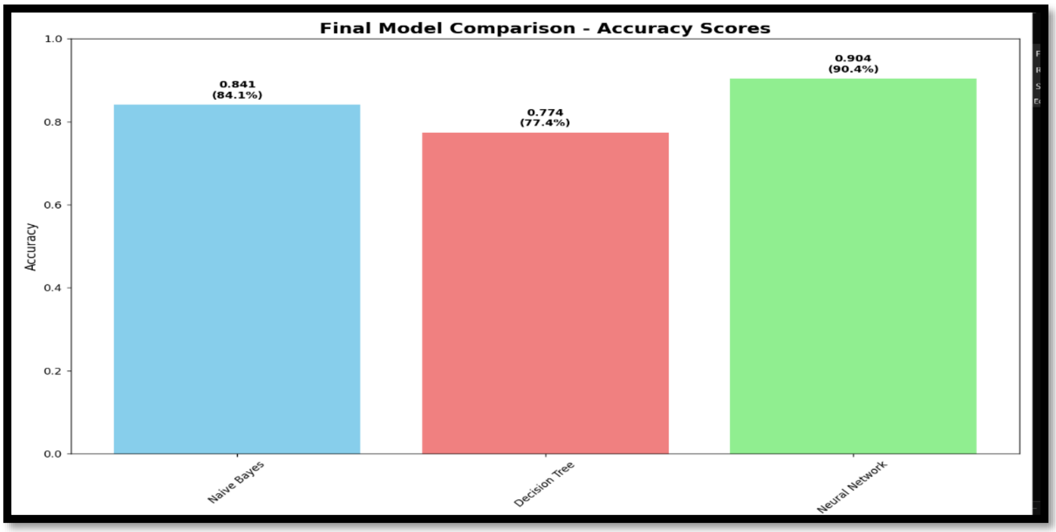

Figure 5 : Models Accuracy

Table 5 : Table For All Result

Classifier	Accuracy	Class	Precision	Recall	F1-score
Naive Bayes	84.10%	Bikes	87.86%	90.42%	89.12%
-	-	Cars	77.45%	78.06%	77.75%
-	-	Planes	87.01%	83.75%	85.35%
Decision Tree	77.41%	Bikes	87.39%	77.92%	82.37%
-	-	Cars	69.05%	77.21%	72.91%
-	-	Planes	77.74%	77.08%	77.41%
Neural Net.	90.38%	Bikes	94.14%	93.75%	93.95%
-	-	Cars	87.02%	87.72%	87.37%
-	-	Planes	89.95%	89.58%	89.76%

This table summarizes the performance of each classification model across the three vehicle classes using accuracy, precision, recall, and F1-score. The Neural Network achieved the highest scores across all metrics and classes, while Naive Bayes showed strong results for Bikes and Planes. The Decision Tree had the lowest overall accuracy and struggled particularly with correctly classifying Cars.

Conclusion:

In this project, we compared the performance of three classification algorithms—Naive Bayes, Decision Tree, and Neural Network on a multiclass vehicle dataset comprising Bikes, Cars, and Planes. The models were evaluated using accuracy, precision, recall, and F1-score, with confusion matrices providing insights into class-wise performance.

The Neural Network achieved the highest accuracy (90.38%) and demonstrated strong performance across all classes, reflecting its strength in modeling complex and nonlinear patterns in the data.

Interestingly, Naive Bayes outperformed the Decision Tree in terms of overall accuracy, despite being a simpler, probabilistic model. This result can be attributed to several factors:

- Naive Bayes tends to perform well when features are conditionally independent given the class, which may roughly hold true for some features in this dataset.
- It is less prone to overfitting compared to Decision Trees, especially in small or moderately sized datasets.
- Decision Trees are sensitive to small variations in the data and can create overly specific rules that don't generalize well, which might explain their confusion between Cars and Planes in this case.

While Neural Networks proved to be the most effective overall, the findings suggest that simpler models like Naive Bayes can still offer surprisingly competitive performance with low computational cost and easier interpretability making them suitable in scenarios where resources or explainability are priorities.

References :

[1] : https://www.kaggle.com/datasets/mohamedmaher5/vehicle-classification