CS401 CA2: Decision Tree and CNN Implementation + Evaluation

Worth 20% of final mark, graded out of 100%

Part 1: Decision Trees – 50%

Choose a dataset that is well suited for classification and that has all numerical features. (e.g. one of the datasets from UCI). Look for a dataset with numerical features and a categorical target variable. If you look at the UCI website you can search for datasets for Classification and numerical and look through what those search filters give you some options. You might want to choose a dataset with less than about 10 attributes to make your decision tree readable. To make the training fast, try to find a dataset with 1000 samples or less.

Rubric

Task 1: Understanding the data – 10%

a. Include a link to and basic description of the data (what the features are).
b. Write your own brief description of what the classification task is and why a decision tree is a reasonable model to try for this data.
c. Include one summary visualization of the data.

Data Understanding	
Description of Dataset	3
Justification of Task	3
Visualisation	4

Task 2: Fit a Decision Tree and evaluate performance – 20%

a. Fit a decision tree with the default parameters on 50% of the data
b. Test it on 50% held out data and generate a classification report
c. Inspect the model by visualizing and interpreting the results
d. Does this model make sense?
e. Are there any leaves that are very small?
f. Is this an interpretable number of levels?
g. Repeat with the **entropy** criterion. Does using the entropy criterion make a big difference or small difference in the overall classifier?
    a. (See the documentation for the sklearn DecisionTreeClassifier. One of the parameters is called criterion)

Basic DT Implementation	
Fit Decision Tree	3
Classification Report 50% Hold Out	4
Visualise and Interpret Results	6
Entropy Criterion	7

Task 3: Experiment with parameters - 20%

Experiment to see how **max_depth, min_values_split, or min_values_leaf** impacts the model

a. Choose one of these and say explain why and how you hypothesise it will impact the performance.
b. Use the model you fit above and EDA (Exploratory Data Analysis) to choose minimum and maximum values for your parameter. Choose a total of 3 values for the parameter, explaining your choice.
c. Retrain the model for each value of the parameter.
d. Test and use at least **3 metrics** to describe the performance, compiling your results into a DataFrame.
e. Plot and interpret your results.

Parameter Experiments	
Parameter choice and hypothesis	3
Parameter value choice and justification	4
Train and test using 3 metrics to describe performance	5
Visualising and interpreting results	8

Part 2: CNNs– 50%

Part 2 of this assignment involves implementing, evaluating and improving a Convolutional Neural Network (CNN) for an object classification task using the CIFAR-10 dataset. You are asked to implement a standard architecture, such as the one covered in class examples and demonstrate an understanding of its performance. Dataset: The CIFAR-10 dataset can be obtained from its official webpage and used for training and testing the models. Optionally, only the first category (1_data_batch) can be used as training data, if resource constraints apply.

Rubric

Task 1: Implementing a standard CNN – 20%

a. Load the CIFAR-10 dataset, give a brief explanation of the features and prepare the data for training (format, normalization).
b. Based on the examination of the data, and CNN architectures discussed in class, propose a CNN architecture appropriate for an object detection task for CIFAR-10.
c. Define and fit the CNN to the dataset, justifying any initial training decisions in a markdown cell e.g., learning rate, optimizer, split.
d. Plot the loss for training and validation and interpret the results.

Implement a Standard CNN	
Data Load + Prep	4
Propose appropriate architecture	4
Train and Test Model	4
Explaining results	8

Task 2: Experiment with architecture and parameters - 30%

Make the appropriate changes to the code, and note the results, with visualisations where appropriate, and explanations in Markdown.

a. Make changes to the number of convolutional layers (e.g., doubling them) and interpret the effect.
b. Change the size of the kernel and explain the results.
c. Choose two suitable alternative values for learning rate, modify the model and interpret the impact of each one.
d. Demonstrate visually and explain what happens if you choose an alternative activation function (ReLU, Tanh).
e. What happens if you use an alternative Optimiser?
f. What impact does batch size make e.g., try batch size of 4, 32 and 256?
g. Visualise and explain the output of each convolutional layer for a sample test image, displaying each channel as a grayscale image.

Experimentation	
Changes to Convolutional Layers	4
Changes to Kernel	4
Changes to Learning Rate	4
Changes to Activation Function	3
Changes to Optimiser	3
Impact of Batch Size	4
Visualisation of convolutional layers	8

What do you need to submit?

You will need to submit 2 python notebooks (Jupyter notebook or collab is fine). You should include CS401 CA2, Name, ID, Programme code in the title markdown cell. Alternate your code blocks with markdown blocks containing your explanations and interpretations of results as well as any justification for decisions made in your experiments, similar to examples done in class. Make sure your submission is clearly labelled with the task number, easy to read and self-contained, i.e. no external files or filename changes are required to run the code. Note marks will be awarded for both code and the explanation in the markdown, it is essential to have both components for each question.

When is it due?

Your submission is due at midnight on **21 Dec 2025.**