6G6Z0048 Artificial Intelligence: 1CWK100

# 1) Longlisting

**Supervised Learning** is the process of training a predictive model as is it given clear instructions on what needs to be learnt and how to learn it.

**unsupervised learning** is the process of training a descriptive model as there is no single feature or particular interest and no target to learn.

**K-Nearest Neighbour** (kNN) is a simple learning algorithm that classifies unlabelled examples using its k-nearest neighbour’s information. Using the features of the examples as coordinates, it plots the examples in a feature space. To classify unlabelled examples, the distance between the example and other labelled examples is calculated using the Euclidean distance method. The unlabelled example is then classified in the same class as the closest labelled example. K is the hyperparameter of the number of neighbours that are compared to the example, this can be altered to improve the accuracy of model. kNN can be applied to image classification. For example, the user uploads an image of a bike, the algorithm will then highlight features of the image, i.e. “wheels” and use the features to compare it to existing labelled images in the feature space. It will then label the new image with its nearest neighbour which is bike.

**k-means clustering** assigns one of the examples to one of the k clusters, where k is a number that has been determined ahead of time. The goal is to minimise the difference in feature values of examples within each cluster and maximise the difference between each cluster. As it is Unsupervised Learning, it starts with an initial guess for cluster assignments and modifies slightly to see for changes that improve homogeneity within clusters. It stops once there is no change is reassignment of examples. In image classification, the images can be assigned to clusters and modified until no changes are made, to train the model. New images can then be classified by the trained model.

**Feature Space Visualisation** is used to represent images as points in a multi-dimensional feature space. Within the feature space lies all possible combination of feature values (Lantz, 2023). Feature space visualisation helps to find patterns and relationships between data, allowing them to be grouped in the similar classes/clusters.

**Hyperparameter Tuning** is the testing of hyperparameter settings to achieve the better model fit. For example, tuning k in kNN to find the best value of k, which is the number of neighbours compared to the unlabelled example.

**Naïve Baye Classification** uses training data to calculate the probability of each outcome based on evidence provided by feature values. It uses calculated probability to predict the class of a new example. Naïve baye assumes that all features in a dataset are equally important and independent (Lantz, 2023). Due to the algorithm using frequency tables to learn the data, each feature must be categorical therefore numerical features must be put into categories known as bins. Naïve Baye requires features to be extracted before the probability can be determined. One approach to this is to use Bag of Visual words, which treats features as words and displays them in a histogram based on how many times it appears. This can be applied to image classification as BoVW would use the image features as words to create a histogram. For example, BoVW would find the words “wheel”, “frame”, “handle”, which are then used by Naïve baye. If a new image had these words, Naïve baye would use the probability to classify this image as bike.

**Bag Of Visual Words** represents data as a binary feature indicating whether each feature appears in an example. It allows images to be directly comparable points in the same feature space.

**Decision Tree Classification** uses a tree structure to model the relationship between features and the possible outcomes. Examples start at the root node and pass through the decision tree based on its features. Each node branches off into more than one node. The example continues through the decision tree until it ends at a leaf/terminal node, which will be the class for the example. The decision tree is a divide and conquer algorithm as it divides the dataset into subsets which are split repeatedly until the process stops as the data in the subsets become homogenous. In Image classification, the example of a bike image will start at the root node of the tree, the first node could be “Does the image have wheels?”. The image will move to the YES node. This will continue until the image has reached the terminal leaf which will classify the image as bike.

**Gradient Boosting** combines the predictions of multiple weak learners sequentially, usually decision trees. It aims to improve overall predictive performance and enhance the model’s accuracy (Saini, 2024). When paired with decision tree, it makes it more accurate with its classification and allows it to handle large datasets like images.

**Ensemble Classification** combines multiple weak learners to create a stronger learner. The ensemble depends on a diverse set of classifiers as their individual predictions are combined into one final prediction. When each classifier makes its own independent prediction, it must do more than simply guess. This means that ensemble has an uncorrelated classification but is better than random chance. Ensemble classification could combine multiple weak image classification learners to create a stronger learner. Each learner would independently decide which class the image should belong to, and the final decision is determined by majority vote.

**Boosting** is an ensemble method, which improves the performance of weak learners to attain the performance of strong learners. Boosting uses ensembles of models trained on resampled data, which vote to determine the final prediction. The prediction is biased as models that perform better have a greater influence over the ensemble’s final prediction. This can improve overall accuracy of image classification as it allows the better performance models to have greater influence over the final decision.

**Bagging** Is an ensemble method which generates new training data using bootstrap sampling on original training data (Lantz, 2023). Datasets are used to generate a set of models using single learning algorithm. Similar to boosting, the models’ predictions are combined via voting for classification. Bagging is often used with unstable learners; these are essential for ensuring diversity as they vary dramatically if minor changes occur in the input data. For image classification, the models are trained on a different subset of images.

**Random Forests** builds upon bagging but adds diversity by allowing algorithms to choose from a randomly selected subset. For example, at the root node it might only be allowed to choose from a small number of features, which are chosen at random from a set of predictors at each split. There is always a random different subset provided so each tree in the ensemble is unique. Similar to bagging, once the ensemble of trees is made, it performs simple votes to make a final prediction. For image classification, each tree will be specialised in different aspects or patterns of data. The relevant features of the image are extracted, and each tree in the forest focuses on a different aspect of the image. Like decision trees, the data will start at the root node and make its way through the tree until a terminal node is reached. Once each tree has made its independent prediction, the overall prediction is the class that was most predicted.

**Logistic Regression** is used to calculate the probability of a binary event occurring (Thanda, 2023). It predicts the probability based on a set of independent variables, which influence the outcome. Logistic regression uses the sigmoid function to model the relationship between the input and probability of outcome. It has a decision boundary that splits the instances of the two classes within the feature space. Examples are classified by which side of the decision boundary they lie upon. In image classification, the features of the image are used as independent variables to find the probability of the class, however, as there are only two classes the image is either true or false. For example, the two classes could be “bike” or “not bike”.

**Multinomial Logistic Regression** is an extension of logistic regression and is used when the outcome being predicted has more than two categories (Libguides: Statistics Resources: Multinomial logistic regression). It uses the SoftMax function to turn a vector of K real values into a vector of K real values that sum to 1 (Wood, 2019). The decision boundary is a hyperplane in the feature space that separates the classes. In image classification, multinomial logistic regression will use the features of the image as independent variables to find where it lies within the feature space, and then calculates the probability of it belonging to the class.

**Artificial Neural Network (ANN)** models the relationship between input and output signals. It is like the human brain’s neural network. ANN uses a network of artificial nodes to solve challenging problems. The activation function is the mechanism by which artificial nodes process incoming information and determine whether to pass the signal to the other nodes in the network. If the activation function threshold is met by the input, then it results in an output signal. Each pixel of an image is treated as a separate feature, and the artificial neural network discovers the relationship between these features to make predictions.

**Convolutional Neural Network (CNN)** is a larger deep learning model with more than one hidden layer. The nodes in one layer are only partially connected to the nodes in the next layer. It is a deep feed-forward network, which is used for visual tasks that independently learn important distinguishing image features. In image classification the convolutional layer extracts the features of an image by scanning through the image with filters (Huellmann, 2022). The Rectified Linear Units (ReLU) layer changes any negative values to zero to introduce non-linearity into the network. The pooling layer downsizes the image to increase computational speed, and the fully connected layer outputs a final classification.

**Feed-forward Network** are networks which feed the input signal continuously in one direction from the input layer to the output layer.

**Transfer Learning** uses existing models that were shared, which are deep learning models that can be adapted from one context to another.

**Fine Tuning** is the additional training that a pretrained neural network undergoes if they do not transfer directly to the new task

**Feature Extraction** is a dimensionality reduction technique that capture the relevant features and highlight patterns.

**Support Vector Machine (SVM)** uses multidimensional surfaces to define relationship between features and outcomes. The goal is to create a flat boundary called hyperplane, which divides space to create homogenous partitions on either side. SVM combines kNN and linear regression. It is adaptable for use with nearly any type of learning task. Support vector provides a compact way to store a classification model. In image classification SVM finds the optimal hyperplane that separates the image classes so, when a new image is added it can be classified by mapping it in the hyperplane, and which sector it is in is the class it belongs to.

# 2) Analysis

For the image categorisation needed by the company, three supervised learning algorithms are the best suited.

The first algorithm to be considered is the gradient boosted decision tree. The decision tree algorithm is simple, efficient, and easy to implement, which corresponds with founder 1's request for a simple approach. However, it is critical to understand that one of the decision tree's limitations is that it may not be the best fit for tasks involving a large number of numeric features, such as images. This limitation derives from the potential for a huge number of decisions and an overly complex tree structure.

This difficulty can be efficiently solved by incorporating gradient boosting into the decision tree design. This dramatically improves the decision tree's performance, allowing it to excel in both classification and numerical predictions. It does this by having multiple decision trees run sequentially and combining their predictions. This guarantees that the algorithm remains adaptable and effective in a variety of settings, fulfilling founder 1's needs for a simple yet effective algorithm. The decision tree is one of the top ten learning algorithms, which perfectly matches with the expectations set by founder 2.

Decision trees have an advantage in terms of interpretability. They can output the resulting structure in a human-readable format, revealing the model's inner workings. This transparency not only helps to comprehend the decision-making process, but it also adheres to the ideals outlined in the 'FAST Track principles', which were adopted by founder 4, who was concerned about AI ethics and legal requirements.

However, it is critical to recognise a potential disadvantage of the gradient boosted decision tree: the presence of multiple hyperparameters that must be fine-tuned to attain the best model fit.

The process of Iterating through numerous combinations for hyperparameter tuning requires significant computer resources, which may be a challenge, especially given founder 3's preference for a solution with minimal computational requirements. In contrast, the decision tree adds to the entire automatic learning process. By following a set of predefined rules, decision trees autonomously make decisions and construct predictive models, reducing the requirement for manual interaction. This meets the preference expressed by founder 3.

While decision trees are extremely effective at building predictive models, they struggle to strike the correct balance between overfitting and underfitting. Overfitting, where the model gets fitted to the training data and fails to generalise, and underfitting, where the model is too basic to capture underlying data patterns, are both inherent dangers. Pruning emerges as a solution, providing a deliberate approach to both difficulties. Pruning improves the model's ability to generalise to new data by systematically reducing the size of the decision trees. A nuanced technique entails allowing the tree to reach its full size before trimming, ensuring that it thoroughly investigates and captures key data structures.

The difficulty of class imbalance provides a significant hurdle for the decision tree learner, as this model may unknowingly acquire bias in favour of the majority classes when certain categories dominate the dataset. Founder 4 has specifically expressed worry over this issue. To address this issue, a recommended technique is to balance the training data before fitting it to the decision tree algorithm.

A significant disadvantage of the decision tree algorithm is its inherent greediness. The algorithm divides the data into partitions one feature at a time, prioritising the identification of the most homogeneous partition at each stage. While this method may speed up the decision-making process, it may limit the algorithm's ability to generate more sophisticated sets of rules. This lack of nuance may lead to inferior performance, preventing the algorithm from reaching higher overall accuracy throughout the entire dataset. As a result, the decision tree's preference for instant uniformity may limit its ability to capture nuanced patterns and nuances in the data, reducing its ability to offer more accurate classifications. However, this is counteracted by gradient boosting, by taking into consideration the previous error of each tree and assigning weights based on it.

The second algorithm being considered is the Support Vector Machine (SVM). The Support Vector Machine is a surface that defines a boundary between points of data plotted in a multidimensional space, representing examples and their feature values (Lantz, 2023). SVM, which combines solutions from linear regression modelling and k-nearest neighbours, could be familiar to Founder 1, who is currently working on a kNN classifier. The combined use of the two techniques provides SVM with the ability to model highly complicated relationships, which is especially useful for image data processing. Given the company's emphasis on object recognition and image categorisation, SVM's inherent ability to handle complicated relationships becomes a valuable tool. The most important challenge in SVM implementation is to discover the maximum margin hyperplane, which optimises the separation of classes. This separation reduces vulnerability to random noise while providing strong classification by keeping each class distinct on its side of the boundary.

Support Vector Machines (SVMs) provide an easier experience than neural networks, thanks to the availability of various well-known SVM methods. This is in line with Founder 3's preference for lowering computational requirements by exploiting existing techniques. It also meets Founder 1's criteria, allowing her to choose between a low-cost or free open-source algorithm. The availability of well-supported SVM alternatives simplifies the deployment process while also catering to the different demands and requirements of the team.

SVM gained popularity due to its amazing accuracy, which is critical in tasks such as image categorisation. In the context of the auction house, where grouping similar items is critical to improving the connection between sellers and buyers, SVM's precision plays an important role. Founder 2's emphasis on accuracy is perfectly aligned with SVM's intrinsic strengths as a powerful learning algorithm. Using SVM, the auction business may efficiently streamline its categorisation process, reducing errors and improving the overall experience for both sellers and buyers.

Resilience to overfitting, a significant issue in machine learning models, is one of Support Vector Machines' inherent strengths. Because SVM uses only a portion of the training data for prediction, it is incredibly resilient. Specifically, SVM reduces the possibility of overfitting its predictions to the details of the training data by concentrating on a chosen subset instead of the full training dataset.

Support vector machines' main flaw is that in order to find the best model, many different parameter and kernel combinations must be tested. It can take a while to complete this procedure, especially if working with datasets that have a lot of features or examples. It produces a complicated black-box model as a result, making interpretation difficult. Given that founder 2 and his team of human moderators would find it challenging to examine the image categorisation carried out by the SVM, Founder 2 would somewhat be concerned about this lack of interpretability. The model's opacity makes it difficult to identify errors, which makes it difficult for moderators to identify misclassifications. Moreover, this intricacy runs against Founder 4's demand for transparency, which affects the algorithm's ability to comply with legal requirements.

Another disadvantage of SVM is that when dealing with huge datasets, such as images, the algorithm takes more processing power, which makes it more expensive. This contradicts founder 3's requirements, as she was concerned about the cost of the cloud-based infrastructure increasing.

SVM can also be memory expensive because it requires more space to store the kernel matrix, which can be immense on larger datasets. This also goes against founder 3's requirements because additional storage would be required, increasing the overall cost.

The third algorithm to be considered is Convolutional Neural Network (CNN). CNN uses numerous layers of data processing because it was designed especially for analysing visual data, including images. CNN uses a multiple layered architecture to enhance its capacity to extract complex features, it also presents interpretability issues that could worry Founder 4 because of legal responsibilities for transparency. CNN's intricate internal mechanisms may make it impossible for Founder 2 and the human moderators to check the algorithms inner workings for image categorisation. Interestingly, CNN doesn’t require human supervision as it is autonomous with steps like feature extraction. Raw pixel data are represented in the input layer, where the process starts. In the convolutional layers, these values are processed for features. These layers create a hierarchical representation of features by applying filters to identify patterns such as edges or textures. In every convolutional operation, activation functions introduce non-linearity. CNN down samples feature maps using pooling layers to control computational complexity. It's crucial to remember that CNN still has a high processing requirement despite this optimisation phase, which goes against Founder 3's intention to lower computational requirements.

2D feature maps are transformed into 1D vectors after going through the pooling layers in order to make processing in fully connected layers easier. The output layer's ultimate prediction is the result of these layers' intricate relationship captured throughout the image. However, training CNN models can take a long time because of their many layers and high computational demand. This could possibly concern Founder 1, who was looking for a straightforward but effective solution. CNN is ranked among the top 10 algorithms, which is consistent with Founder 2's focus on accuracy in image recognition and classification. This fits the overall company’s requirement as it is highly accurate at image recognition and classification.

Convolutional Neural Networks demonstrate impressive capabilities when handling large datasets; yet their performance is contingent upon large amounts of labelled data in the training phase. The significant processing power needed is a result of the dual requirements of managing enormous datasets and the requirement for a large amount of labelled data.

Because convolutional neural networks use a lot of resources and have an intrinsic computational complexity, they may execute at a slower pace. Deep architectures have many layers and parameters, which add to their computational load and need a significant amount of computing power for both training and inference. Complicated mathematical procedures like convolutions, larger model sizes, and greater input resolutions can all add to the slower execution.

# 3) Recommendation

In summary, I highly recommend employing the gradient boosted decision tree learning algorithm for image categorisation within the company. This choice is grounded in its ease of implementation and operational efficiency. Considering Founder 1's lack of prior experience in AI/ML software, it is crucial to opt for a straightforward algorithm that facilitates seamless integration. The company's anticipated growth suggests the potential addition of new developers who may collaborate with Founder 1 in implementing the learning algorithm. Therefore, the chosen algorithm must not only be efficient but also easily understandable for new recruits. Currently, Founder 2 oversees a team of human moderators responsible for reviewing and correcting item categorisation. To ensure a smooth transition to automated categorisation, the algorithm must align with the existing human processes. The decision tree, known for presenting its outputs in a human-readable format, stands out as an ideal choice. This feature facilitates the human moderators' ability to verify the algorithm's categorisation accuracy and promptly identify any potential issues within the decision tree, thus streamlining the integration of AI technology into the company's workflow. The company leverages an Infrastructure as a Service (laaS) Supplier, a cloud-based infrastructure that stores the system’s backend. The cloud-based nature of this service affords the company the flexibility to dynamically adjust storage capacity, network bandwidth, and processing power in response to evolving requirements, all with a cost. Founder 3 has expressed a particular concern about costs, making it a critical consideration in the learning algorithm decision. In this context, the decision tree algorithm emerges as a favourable choice, given its relatively economical nature. Decision tree requires minimal financial investment., which aligns well with founder 3’s cost-conscious approach. Being a UK-based company, strict adherence to local regulations is paramount to maintain legal standards. Given the nature of the company as an auction house, there may be a need to subject the learning algorithm to legal inspection. Considering this requirement, the decision tree algorithm stands out as the most suitable choice. Its unique ability to present results in a human-readable format not only facilitates transparency but also aligns seamlessly with the legal review process.

Although gradient boosted decision tree is the better suited learning algorithm for the company due to the founders’ requirements, it isn’t the best algorithm for image categorisation. It is susceptible to noisy data which reduces the generalisation of the model making it unable to correctly classify unseen data. Although the gradient boosting improved the decision tree’s handling of large data, it is still unable to handle extremely large data, so it is limited.

# Reference list

Huellmann, T. (2022) *Image processing: How do image classifiers work?*, *RSS*. Available at: https://levity.ai/blog/how-do-image-classifiers-work#:~:text=The%20four%20types%20of%20CNN,through%20the%20image%20with%20filters. (Accessed: 17 January 2024).

Lantz, B. (2023) Machine learning with r: Learn techniques for building and improving machine learning models, from data preparation to model tuning, evaluation, and working with Big Data. Birmingham, UK: Packt Publishing.

*Libguides: Statistics Resources: Multinomial logistic regression* (no date) *Multinomial Logistic Regression - Statistics Resources - LibGuides at Northcentral University*. Available at: https://resources.nu.edu/statsresources/Multinomiallogistic (Accessed: 17 January 2024).

Saini, A. (2024) *Gradient boosting algorithm: A complete guide for beginners*, *Analytics Vidhya*. Available at: https://www.analyticsvidhya.com/blog/2021/09/gradient-boosting-algorithm-a-complete-guide-for-beginners/ (Accessed: 17 January 2024).

Thanda, A. (2023) *What is logistic regression? A beginner’s guide [2023]*, *CareerFoundry*. Available at: https://careerfoundry.com/en/blog/data-analytics/what-is-logistic-regression/ (Accessed: 16 January 2024).

Wood, T. (2019) *Softmax function*, *DeepAI*. Available at: https://deepai.org/machine-learning-glossary-and-terms/softmax-layer (Accessed: 17 January 2024).

# Appendix 1: LLM conversations

* How is gradient boosting used in decision tree

ChatGPT

Gradient Boosting is a machine learning technique that can be used with decision trees to improve their performance. The basic idea behind gradient boosting is to train a sequence of weak learners, typically decision trees, and combine their predictions to create a strong learner. Here's a step-by-step explanation of how gradient boosting is used with decision trees:

Initial Decision Tree:

The process begins with a simple model, often a shallow decision tree. This tree is called a weak learner because it alone might not perform well on the given task.

Error Calculation:

The errors or residuals from the initial model are calculated. The residuals represent the difference between the actual and predicted values.

Build a New Decision Tree:

A new decision tree is constructed to predict the residuals (errors) from the initial model. This tree is trained to capture the patterns in the errors.

Combine Predictions:

The predictions from the new tree are combined with the predictions from the initial model. This combination is done by updating the predictions of the initial model with the predictions from the new tree.

Repeat:

Steps 2-4 are repeated for a specified number of iterations or until a stopping criterion is met. Each new tree is trained to predict the residuals from the combined predictions of the previous trees.

Final Prediction:

The final model is the sum of all the weak learners' predictions. This aggregated model tends to perform better than any individual weak learner.

The "gradient" in gradient boosting refers to the optimization process used to minimize the errors. It adjusts the model's parameters (in this case, the parameters of the decision trees) in the direction that minimizes the error. The term "boosting" refers to the iterative process of combining weak learners to create a strong learner.

The use of gradient boosting with decision trees provides several advantages, including improved predictive performance and the ability to handle complex relationships in the data. However, as mentioned in the passage you provided, one drawback is the potential complexity introduced by the need to tune hyperparameters, and the computational requirements can be substantial.

# Appendix 2: Source code

There is no source code to submit.