University of Bristol – MSc Data Science (PGT)
EMATM0067: Introduction to Artificial Intelligence Coursework

Student ID: 2623567

1 Question 1: Unsupervised Learning

1.1 [m1] Data Acquisition, Data Cleaning and Preprocessing: (Code cells – 1-6)

In the Natural Language processing (NLP) pipeline, the algorithm has several steps such as pre-processing, extracting word features, extracting sentence-level features, and performing classification, sequence labelling, and topic modelling [1], [2]. Before concatenating the books *Little Women* by Louisa May Alcott and *The Blue Castle* by L. M. Montgomery, cleaning each individually helps retain only the story content, which is richer in context and suitable for clustering tasks in computational linguistics [3].

Code Cell 1-2: Data Cleaning: We removed the start and end markers to acquire the body text data. This step allows downstream NLP tasks to focus on the relevant content, giving us more useful features for a classifier to perform NLP tasks and distance metrics analysis. [1]

We have incorporated steps such as line normalisation [1] by removing the blank lines and stripping whitespaces. Word or token statistics [1] to count the words in each book to ensure the balance between the bag of words in the books. The word count showed balanced lengths between the books, The Blue Castle with 68k words, and Little Women with 192k words.

Code Cell 3-4: Sentence Segmentation, Token Normalisation and Rare Word Removal: To enable sentence-level similarity analysis, we performed sentence segmentation using the nltk library's `sent_tokenize()` method [14]. Also calculated the number of sentences, total words, and minimum and maximum sentence lengths, which are the baseline steps of the test preprocessing before word tokenising. This exploration gives us an idea that both books have long and diverse sentence structures, which helps in building strong word co-occurrence relationships for the similarity calculations later [14]. After sentence-level cleaning, the two cleaned novels were merged into a single dataset, and the text was converted to lowercase. We have removed common high-frequency function words or stopwords like "the," "to," and "of" using NLTK's `stopwords.words()` [6], [14]. Hence, the tokens are reduced from 128701 to 122223 after removing rare occurring words, with 7973 unique words remaining.

Code Cell 5: Lemmatisation: Each sentence is then tokenised and processed using SpaCy's lemmatiser, which reduces tokens to their base or root forms. After lemmatisation, the text contains 135,268 tokens with 14,351 sentences and 9,946 unique or distinct lemmatised words [2], [14].

Code Cell 6: Named Entity Recognition (NER) and Character Filtering: In the books, character names are dominating the vocabulary due to their high frequency. To remove the unnecessary dominating tokens, we used SpaCy's Named Entity Recognition and extracted all "Person" entities [2], [15]. A custom stoplist was then created for names which occurred more than or equal to 5, which helped to remove 104 high-frequency character names to ensure the model's balance during training.

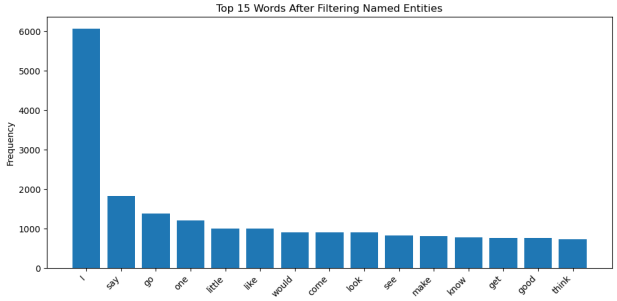

Fig. 1: Words after Filtering Named Entities Using SpaCy

1.2 [m2] Vocabulary Building (POS+Freq and LDA Topic Word Lists)

Code Cell 7: POS-Based Vocabulary Filtering with Named Entity Exclusion: We are further using Part-of-Speech Tagging, which is a sequence labelling that refers to the syntactic role which assigns each word in the text to NOUN, AUX, VERB, DET and NOUN [1], [2]. This is another text normalisation step to construct a clean, interpretable vocabulary for clustering steps [2]. We have used SpaCy's POS tagger on the merged book text and selected only the tokens tagged as NOUN, ADJ, or ADV [1], [2], [15]. By the removal of action-specific words, such as verbs, for semantic clustering. For each POS group: [2].

Code Cell 8: POS + Frequency Balanced Vocabulary (Option A): Using the Part-of-speech tagging [2], we categorised a split with 50 nouns, 25 adjectives, and 25 adverbs for the 100 content-rich words or tokens required for our question. To have the best quality dataset from both books, the tokens are drawn from the top 200 most frequent words from each category. We also did `random.seed(42)` to ensure reproducibility. The final list of words is saved in the `final_outputa.txt`. This balanced approach prioritises interpretability, contextual coverage, and semantic richness.

Code Cell 9: Topic-Based Vocabulary Selection (Option B – LDA): In Option B, we created another list of 100 words based on semantic themes using Latent Dirichlet Allocation (LDA) [18]. LDA is a generative model which views every text corpus as a latent structure [3], [18]. Using the Gensim library, we filtered infrequent and overly common terms and trained an LDA model on the dataset. After training a 7-topic LDA model, each topic was analysed to uncover its latent theme. Topics ranged from literary and narrative themes (Topic

0: writing & storytelling) to interpersonal and emotional dynamics (Topic 1: family, Topic 4: relationships), practical movement (Topic 2: journeys), dialogue and reflection (Topic 3), and visual description (Topic 6) [3].

Code Cell 10: Venn Diagram – to compare Option A and Option B: We plotted a Venn diagram to check the divergence between the two lists. Only 5 words overlapped between the two, indicating a 95% divergence.

1.3 [m3] Co-occurrence Matrix Construction & Similarity Scores – Code Cells 11–13, 29, 32

Code Cell 11: Co-occurrence Matrix Construction: To perform [m3] and find the word similarities, co-occurrences and distance between the words, we have constructed the co-occurrence matrices from both Option A and Option B lists. Using the context window of ±4 tokens, each word in the list is compared to its neighbouring words in the lemmatised tokens [1].

The function `build_cooccurrence_matrix()` is used to count the co-occurring token word pairs and store the result in a square matrix dataframe (100x100). These matrices were saved as `cooccurrence_matrix_optiona.csv` and `cooccurrence_matrix_optionb.csv` files. This co-occurrence approach is fundamental to distributional semantics, based on the idea that words appearing in similar contexts tend to have similar meanings [2], [16].

Code Cell 12: Interpreting the Co-occurrence Heatmaps:

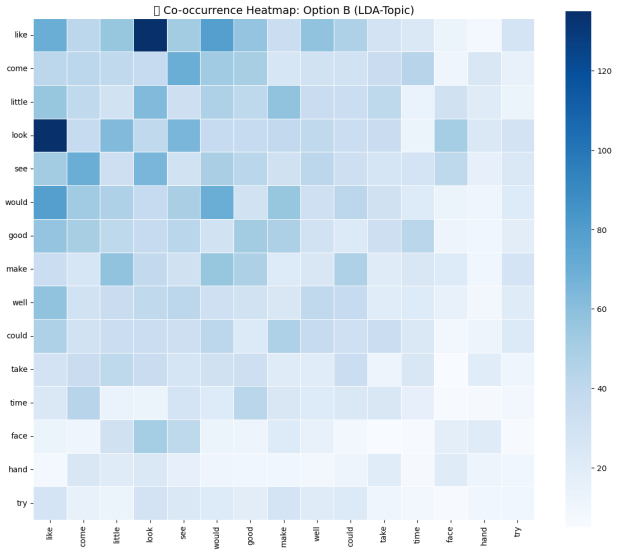

Fig. 2: Co-occurrence Heatmap of Top 15 Tokens for Option B list - LDA topic modelling

Visually representing the co-occurrence matrix in a heatmap with `top_n = 15`, we can observe that in Option A list, which is the grammatical or semantic words chosen based on the POS tagger and frequency, the clusters are observed to be sparse and moderate interaction among context-driven words like *home*, *face*, *way*, and *change* shows the lower intensity and sparse clustering, reflecting on the list's linguistic diversity.

In comparison, the heatmap of the Option B list, which contained the topic-modelled or thematic words, shows more dense clustering and words like *look*, *come*, *would*, *make*, and *good* have strong co-occurrence. This observation gives us an idea of how Option B captures narrative-driven semantic clusters. **Code Cell 13: Top Co-occurring Word Pairs**: We have printed the highest and lowest co-occurring word pairs in the Option A list so we can interpret the distance metrics better.

1.4 [m4] Distance Matrix Transformation & Clustering (14–17, 26–28)

Code Cell 14-15: Co-occurrence Distance Comparison, PCA Embedding, and Dimensionality Reduction: To regularise and normalise the word frequency vectors, which are the similarity-based co-occurrence data, using row-wise L2 normalisation using sklearn.preprocessing() [4], [6]. We then compute the cosine distance matrix, which is given in the formula below: *Cosine distance = 1 - cosine similarity*

In this, the smaller the values, gives higher the contextual similarity between words [4].

	Top A Co-occurring Pairs	Top B Co-occurring Pairs
0	much & one (37)	like & look (135)
1	one & much (37)	look & like (135)
2	one & mother (29)	like & would (79)
3	mother & one (29)	would & like (79)
4	way & one (26)	come & see (70)

Fig. 3: Top co-occuring word pairs in Option A and B

After calculating the cosine distance of the word vectors, we calculate the Principal Component Analysis (PCA). PCA is a dimensionality reduction algorithm using Scikit-Learn's PCA performs under the assumption that the data is centred at an origin [4]. We have used `n_components=2`, which gives us a 2-D representation by dimensionally reducing the 100x100 matrix helps us with data reduction and feature extraction [4], [12]. We displayed the top 5 co-occurring pairs using vector-normalised tokens to see their effects after the PCA dimensionality reductions were applied. We plotted and visualised the 2-D PCA for both the option A and option B lists. In PCA projection of POS and frequency, we can see semantically related words appear closer together (e.g., home–place, mother–child, look–see). This confirmed that our preprocessing pipeline preserved meaningful distributional patterns [1], [2]. Option A scatter plot has more scattered visualisation, whereas Option B has dense, grouped clusters showing tight thematic or topic grouping.

Code Cell 16 - Reading Top Words Pairing: In order to compare the two vocabulary lists strategies, we considered only the top 10 most commonly co-occurring word pairs from both Option A (POS tagging) and Option B (LDA Topic Modelling). Option A word pairs emotional and family co-occurrences (e.g., mother & one, home & mother, eye & one), while Option B shows dialogue and action-related co-occurrences (e.g., look & like, see & come, would & like).

Code Cell 17: Co-occurrence Network Graphs: We have also plotted the top 10 pairs Correlation network, which is a co-occurrence graph, to see the relationships between the pairs

[18]. In the Option A list, ”one”, ”mother”, and ”home” are observed to be the centres, and in the Option B list, ”look”, ”like”, ”see”, and ”make” are most connected.

Code Cell 26-28 – K-Means Clustering and Visualising Clustered Words in a Table: We performed the unsupervised learning K-Means clustering, projecting the Dijkstra-based distance matrix using PCA with $k = 3$, which is determined by the Elbow method, which we will be defining in m5 and 3 labelled groupings help us obtain these groupings of the vector tokens below:

For Option A list:

- Cluster 1 included perceptual and emotional terms like face, eye, and mother
- Cluster 2 grouped spatial tokens like minute, home, turn
- Cluster 3 reflected more abstract or descriptive words like death, stupid, creature

Whereas in the Option B list:

- Cluster 1 narrative and emotional words such as “whisper,” “gentleman,” “follow,” “door,” “trouble,” “manner.”
- Cluster 2 includes words like “could,” “well,” “would,” “make,” “see,” “mine,” “also.” spoken language or modal verbs or stylistic elements.
- Cluster 3 words relate to action, such as “take,” “away,” “back,” “find,” “way.”

1.5 [m5] Dijkstra-Based Graph Distance & Advanced Clustering (Code Cells 18–25)

In m5, we explore the graph-based semantic modelling and clustering techniques to understand and observe the lexical structures within both POS-filtered (Option A) and LDA-topic-based (Option B) words/tokens. By incorporating Dijkstra’s algorithm [2] and dimensionality reduction [4], this phase extends unsupervised learning beyond local co-occurrence patterns and captures indirect semantic relationships across the text corpus of the books/novels merged.

Code Cell 18 – Hierarchical Clustering (Option A): K-means clustering is also known as K- Medoids which requires the number of clusters to be defined for the algorithm where Why are performing hierarchical clustering is in this method there is no such requirement and we can observe the 2 D linkages and observe the dissimilarities between the co-occurrence words without the k no of clusters constraints [5].

To derive the hierarchical clustering with the Option A list of words, we used the square and symmetric co-occurrence matrix of the Option A list and calculated the co-occurrence to distance by the formula.

$$\text{Distance}_{ij} = \frac{1}{\text{Co-occurrence}_{ij} + \epsilon} \tag{1}$$

Following the unsupervised learning practices outlined by Géron [4], we converted the co-occurrence similarity matrix into a distance matrix using the inverse transformation [4], [6] to enable graph-based modelling and shortest-path clustering techniques.

Performing agglomerative hierarchical clustering [5] with Ward’s linkage [5], [7], we constructed a dendrogram to visualise usage similarity [5]. The algorithm generates a tree linkage by iteratively merging the pair of clusters whose reduction in total within-cluster variance is smallest. Words like ”mother,” ”home,” and ”child” formed close interlinked branches and Emotional words and location nouns were more likely to appear at the far ends, testifying to unambiguous substructures in the corpus [5], [3], [18].

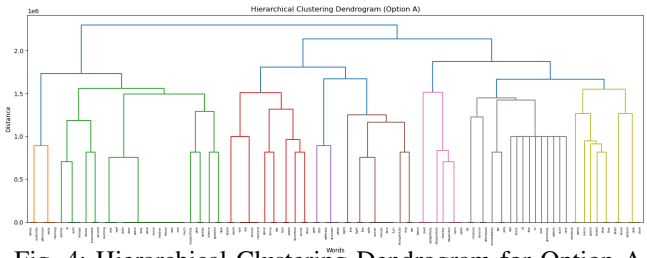

Fig. 4: Hierarchical Clustering Dendrogram for Option A

Code Cell 19 – Hierarchical Clustering (Option B): The hierarchical clustering of Option B shows clear semantic clustering between LDA-selected words, with word clusters around themes of family, emotions, and time. The structure indicates strong contextual relationships, indicating LDA topic modelling causes more semantically meaningful vocabulary clusters. This enhances the performance of Option B in detecting intent-based, theme-based word clusters.

Code Cell 20 – PCA + K-Means Clustering (k=5) – Option A: To observe the flat clustering of the Lists, we just used the value for the number of clustering groups as K=5 based on initial visual cues and heatmap patterns to visualise the clustering in 2D using PCA on the co-occurrence matrix of the Option A list. We have distinct and slightly overlapping clusters. One cluster grouped relational nouns like mother, child, home and another grouped perceptual descriptors like eye, face and Emotional or abstract terms like death, creature clustered together [4], [6].

Code Cell 21 – PCA + K-Means Clustering (k=5) – Option B: Similarly, we used k = 5 for the K-means clustering with PCA 2D reduction using the co-occurrence matrix for the Option B list and observed that there are distinct groupings of the clusters. While less geometrically distinct, clusters emerged along expected lines: words like - “look,” “would,” “see,” “like” formed an action/interaction group and “character,” “daughter,” “professor” clustered as narrative nouns and “comma,” “quote,” “period” clustered as formatting-related items [4], [6].

Code Cell 22-23 – Elbow Method and K-means Clustering (k=3) (Option A): As we mentioned before in the Flat Clustering of the K-means, and used the value of k =5 to observe the clustering. We are using the Elbow rule method [4], which is used to determine the optimal number of clusters for a dataset. We used the elbow method and plotted the graph to find the optimal number for Option A, and we can clearly interpret that k = 3 is optimal, as increasing cluster count beyond this yields diminishing returns.

The elbow method is a statistically correct interpretation for selecting the number of groups for the list [4], [6].

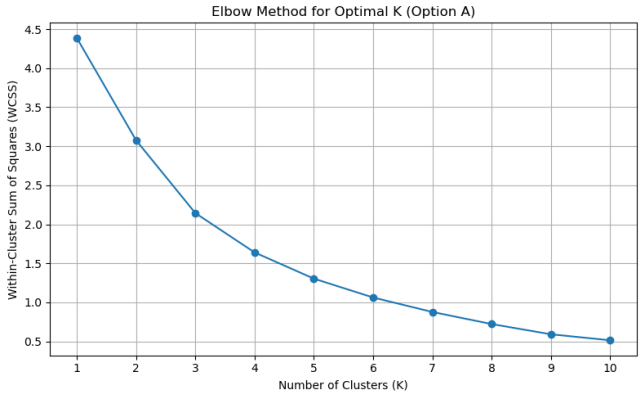

Fig. 5: Elbow Method Plot for POS list - optimal k=3

Using k =3 and plotting the PCA projection. The clusters were semantically coherent:

- Cluster 1: Visual and perceptual terms (face, eye, look)
- Cluster 2: Temporal/spatial tokens (minute, home, turn)
- Cluster 3: Emotional/abstract terms (death, creature, stupid)

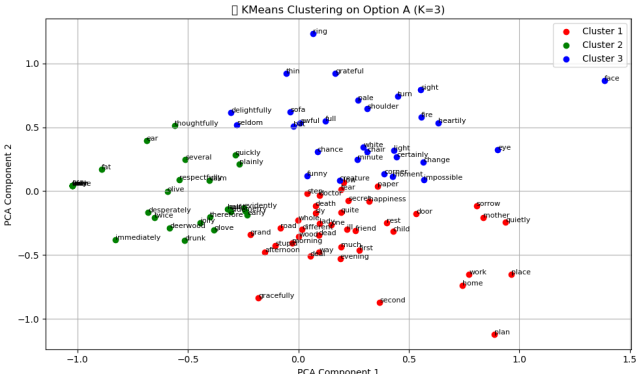

Fig. 6: PCA K-Mean clustering for Option A with optimal k=3

We can observe distinct groups and no evident overlapping between groups. This step confirmed that PCA with K-Means using the elbow rule could effectively extract meaningful semantic structures even from POS-filtered word sets [4].

Code Cell 24 – Code Cell 24-25 Elbow Method (Option B): Similar to Option A, the same elbow plot was applied to Option B. Despite higher variability due to LDA-topic overlap, the optimal k=3 shows a good balance between cohesion and separation.

The resulting clustering groups after plotting the PCA reduced K-Means Clustering using the k = 3 we have:

- Cluster 1: Verb/action terms (look, make, see)
- Cluster 2: Structural and formatting tokens (comma, quote, period)
- Cluster 3: Narrative subjects (character, daughter, professor)

These groupings aligned well with LDA topics and highlighted the model's ability to surface thematic patterns embedded in the narrative flow of both novels.

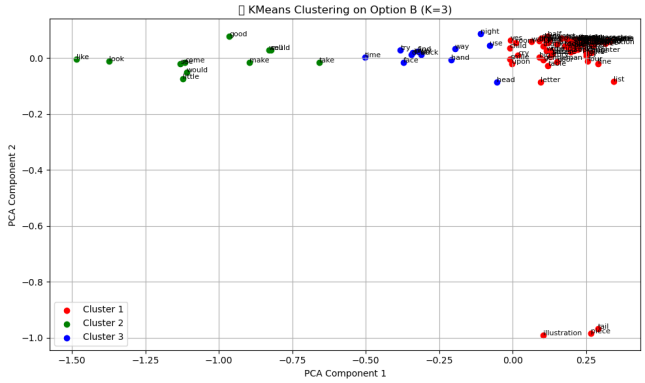

Fig. 7: Elbow Method Plot for Option B showing optimal k=3

	Word 1	Word 2	Co-occurrences	Avg Distance	Similarity Score
3962	one	much	37	0.63	58.730159
3014	mother	one	29	0.71	40.845070
2277	way	one	26	0.74	35.135135
2609	home	mother	20	0.80	25.000000
1077	minute	one	19	0.81	23.456790
...	...	...	...	...	...
2270	way	half	5	0.95	5.263158
2892	place	much	5	0.95	5.263158
4108	much	first	5	0.95	5.263158
728	face	light	5	0.95	5.263158
3041	mother	lady	5	0.95	5.263158

Fig. 8: Top-ranked semantic word pairs with co-occurrence, Dijkstra distance, and similarity score.

1.6 [m6] Evaluation & Interpretation of Clusters and Word Pair Relationships (Code Cells 29–32)

Code Cell 29–30: Dijkstra Distance Matrix, PCA + KMeans Clustering (Option A): We finally normalised and covered the co-occurrence matrix to distance and built a graph for Dijkstra's and calculated the shortest distance between the pairs. The Dijkstra matrix was projected into 2D using PCA with K-Means (k=3) clustering, and we can observe that the words like "mother–home", "one–much", and "much–home" are also closer visually in 2D projection.

Code Cell 31: MDS + KMeans Clustering (Dijkstra – Option B): For Option B, we used MDS to preserve the pairwise distances or dissimilarities of the Dijkstra matrix. K-Means with the clusters thematically with terms (professor, splendid), narrative subjects (daughter, character), and formatting tokens (quote, comma) appearing visually together and also in the same clusters [3], [5].

Code Cell 32: Semantic Similarity Scoring: In m6, we calculated the semantic similarity scores for unsupervised learning, combining co-occurrence word pair frequency with Dijkstra's distance to find and interpret if the frequently co-occurring word pairs that are also closely connected in the semantic graph receive higher scores, capturing both local frequency and semantic closeness.

$$\text{Similarity Score}_{ij} = \frac{\text{Co-occurrence}_{ij}}{\text{Dijkstra Distance}_{ij}}$$

The semantically similar word pairs in Option A: The word "one" is involved in multiple semantically similar word pairs.

It acts as a syntactic bridge across multiple meanings and topics, showing both centrality in usage and thematic overlap.

The semantically similar word pairs in Option B: These word pairs look-like, see–come, and would–like, show the frequent and meaningful co-occurrences in clusters. The low Dijkstra distances show tight semantic coupling in LDA-generated narrative word-pairs.

2 Question 2: Supervised Learning

In this Question, we implement the supervised learning models to handle the classification of the generated 2-D dataset with a curved decision boundary, which is defined by the equation: $y = ax^2 + x + x$. We performed and evaluated the performance of logistic regression and neural networks of increasing complexity (single-layer perceptron, small MLP, deep MLP, and wide MLP) to understand and compare the performance between models to give a better analysis [4], [5].

2.1 Dataset Generation and Decision Boundary

We generated the dataset by using the formula – $Y = aX^2 + x$, which is a quadratic function with a base parameter $a = 0.5$, which controls the curvature of the decision boundary in the dataset. We added a random seed for reproducibility and also used the feature space of $[-4, 4]$ to ensure uniformity across.[4], [6]

Plotted the decision boundary of the data distribution with an initial value of the number of data points as $n = 250$, which is as figure below, giving us a parabola with class A (aquamarine) above the boundary line and class B (blue) and the decision boundary line (red) below.

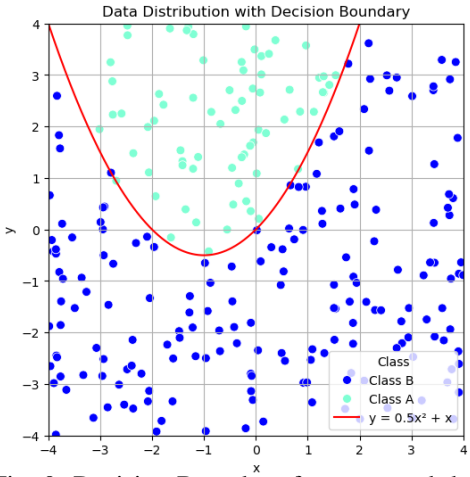

Fig. 9: Decision Boundary for generated data

2.2 Logistic Regression Training

Logistic regression is one of the most common linear classification supervised machine learning algorithms and a discriminative classifier which distinguishes the classes [1], [6]. We split the data into train and test, and to fit the non-linear dataset on a linear classifier, we used a polynomial features pipeline with degree 2 to expand the feature space. Fit the model on the training data to predict the test data. With a =0.5 and a number of points of 250, we got a solid accuracy of 96%. To visualise and interpret the true positives, false positives, true negatives, and false negatives, we plotted the Confusion matrix below. With the False Positives 2 and False Negatives 1, we observe that the model is generalised well. [4], [6], [11]

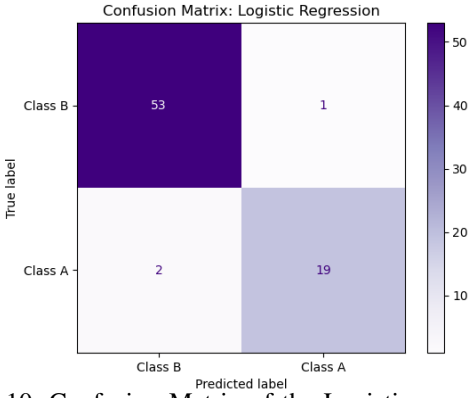

Fig. 10: Confusion Matrix of the Logistic regression

We also plotted the decision boundary of the Logistic regression model, the boundary line $0.5x^2+x$, which is plotted with the dashed line, smoothly separates the classes A and B, and shows us that the model achieved a high-quality fit. [4], [6]

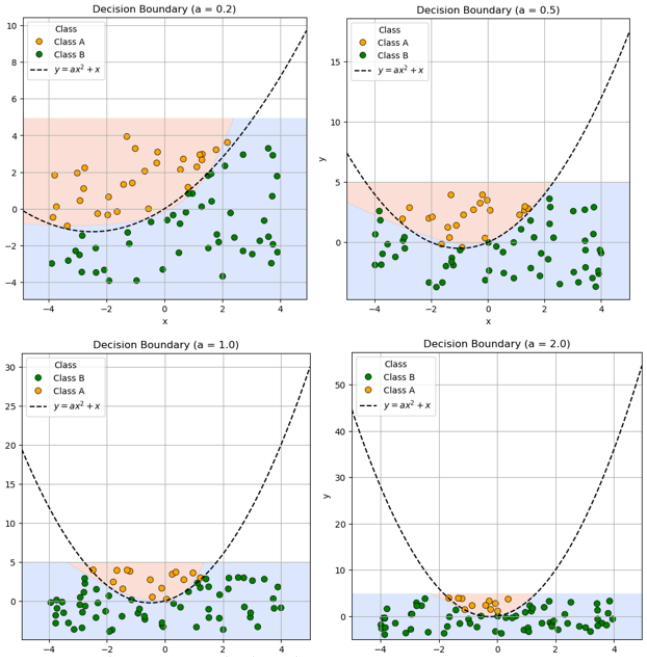

Fig. 11: Variation in Curvature parameter a

We now evaluated the model performance by varying the Curvature parameter with values a = [0.0, 0.2, 0.5, 1.0, 2.0] to check the robustness and limitations of the model. The model shows that the accuracy is consistently high across all curvature levels, and the boundary becomes more curved and steeper. Whereas, generally, as the curvature parameter is increased, the accuracy is meant to decrease, but the polynomial transformation allowed the model to approximate the true boundary effectively. [4], [5], [6], [9]

To check the model's performance in realistic data perturbations, we increased the size of the data set by increasing the data points. We also induced class imbalance and added Gaussian noise to test the model. The stability across all the cases with as high as greater than or equal to 96% even with noise and imbalance in the dataset. Performance is increasing with more data. Even though high accuracy is observed, the model predictions underrepresent the minority class (Class A) when imbalance is introduced. For example, in the 250-point dataset, TP drops from 19 to just 6. [4], [5], [6], [9]

2.3 Single Layer Perceptron

To set a base for comparison with a simpler model for neural networks, we implemented a Single Layer Perceptron (SLP), a linear classifier with no hidden layers [6]. We plotted the confusion matrix and the decision boundary for the value a=0.5, similar to how we assessed the linear regression model. The SLP struggled to learn from the curved decision boundary, and in the plot, we observe a very poor capture of the data points [7].

The model's performance was lower compared to the logistic regression, with test accuracies ranging from 80% to 85%, particularly dropping under noisy or imbalanced scenarios [4]. The SLP with no hidden layers and fitting the data with an accuracy of 80–85% still provides a useful benchmark and justifies the need for deeper, non-linear models explored in the following sections [9], [10].

	n_points	imbalance	noise	accuracy	true_negatives	false_positives	false_negatives	true_positives
0	250	False	False	0.960000	53	1	2	19
1	250	False	True	0.960000	53	1	2	19
2	250	True	False	0.967213	53	1	1	6
3	250	True	True	0.983607	54	0	1	6
4	500	False	False	0.993333	108	0	1	41
5	500	False	True	0.986667	107	1	1	41
6	500	True	False	0.983607	108	0	2	12
7	500	True	True	0.991803	107	1	0	14
8	1000	False	False	0.986667	214	3	1	82
9	1000	False	True	0.990000	216	1	2	81
10	1000	True	False	0.991837	217	0	2	26
11	1000	True	True	0.987755	217	0	3	25

Fig. 12: Confusion Matrix of the Logistic regression

2.4 Small MLP (1 Hidden Layer)

We next implemented the Small Multi-Layer Perceptron (MLP) with one hidden layer with 8 neurons and ReLU activation function used on the dataset. Compared to the single-layer perceptron, this model can train the nonlinear data, which is effective for the task. Additionally, we used Adam optimizer which is used to fine tune features with maximum of 1000 iteration to help the model converge well in noisy and imbalanced datasets We use this MLP to develop a better performance compared to simpler models like logistic regression or the Single Layer Perceptron (SLP).[9], [15], [7], [1]

We plotted the Confusion Matrix for the Small MLP, where the classification shows strong performance of 105 true negatives and 40 true positives with an accuracy of 96.67%. The decision boundary of the model shows that the curved separating region learned by the model aligns well with the true boundary between the classes.

Compared to the straight-line boundary of a single-layer Perceptron, we observe with small mlp how hidden layers allow the MLP to adapt the non-linear decision boundary, which is crucial for accurately classifying data generated by a quadratic function like $y = ax^2 + x$.

We implemented the increase in the value of a in Small MLP, which shows us that with an increase, the true boundary becomes progressively more nonlinear. In the plot below, the colours orange and blue show the class separation that the MLP learned, which is the MLP's boundary line, whereas in comparison, the dotted line shows the true boundary line. We can see how accurately the model fits the datapoints. The accuracy neither increases nor decreases, and the mean of the accuracy 98%. The model is predicting the classes through sharp bends, which shows underfitting in the data. [6], [4], [12] We again interpreted the model's performance by inducing the varying dataset sizes, class imbalance, and Gaussian noise. Across all the scenarios with the highest accuracy, with 500 datapoints and with the class imbalance and noise, shows the model's capacity to classify the data well. The small MLP therefore, is outperforming the simpler logistic regression model.

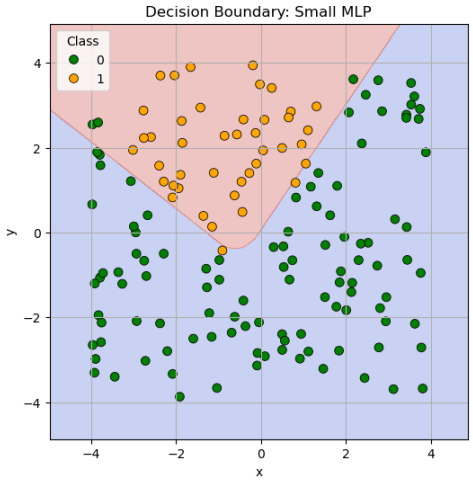

Fig. 13: decision boundary of small MLP

2.5 DEEP MLP

We implement more complexity for the MLP neural network, which is a deep MLP which has 2 hidden layers, the first hidden layer with 8 neurons and the second with 4 neurons. In comparison to the small MLP with a single hidden layer, this architecture increases the depth to learn higher-level features to classify the non-linear data with successive transformations [5], [6], [7]. The model achieved a 97.3% accuracy, the highest among the other models implemented and with minimal misclassification observed in the confusion matrix.

As the parameter a increases, the accuracy increases steadily. With the 12 scenarios, the Deep MLP is giving similar modelling to a single-layer perceptron on the small data (n = 250) with imbalance and noise. [5], [6]

2.6 WIDE MLP

The Wide MLP architecture with a single hidden layer with 64 neurons, increasing the complexity and expanding the model's capacity by increasing the number of units per layer rather than the depth. This helps to capture parallel transformations. This has a similar accuracy of 96.67 % like the Small MLP. It consistently approximates the nonlinear true boundary and exhibits robustness to noise and class imbalance, and shows a steady increase in the model performance with an increase in the number of data points. [5], [6], [7], [11]

2.7 Conclusion

	Model	Accuracy at a = 0.5
0	Deep MLP	0.973
1	Small MLP	0.967
2	Wide MLP	0.967
3	Logistic Regression	0.960
4	Single Layer Perceptron (SLP)	0.800

Fig. 14: Variation in Noise and class imbalance in small MLP

The accuracy of all the models at a curvature sensitivity parameter $a = 0.5$ is shown in the table above. The Deep MLP model had the highest accuracy at 97.3%, followed by Small MLP and Wide MLP, both models at an accuracy of 0.967%. Logistic Regression is also similar, with an accuracy of 96%, while the Single Layer Perceptron lagged far behind at 80%. Deep MLP was strongest among all 12 scenarios with class imbalance and the added Gaussian noise. Logistic regression was good with slight variation in the 12 scenarios. Small MLP performed better in the classification learning curve, even with the presence of noise and imbalances across increasing dataset sizes. In general, as the model depth increases in neural networks, enhanced performance compared to smaller, simpler models, with Deep MLP doing slightly better generalisation at larger dataset sizes.

3 Question 3: Essay

A Large Language Model is a type of Artificial Intelligence, a machine learning model that is pre-trained with a natural language or human language corpus, which generates human-like text. An LLM is a tool developed to interact with humans when prompted. The attributes of personhood are characteristics such as human nature, the ability to act independently, self-awareness, an understanding of time, and the recognition of rights and responsibilities, among other traits. LLMs' rights or attributes to personhood are a thought-provoking, diplomatic, and contradictory statement, yet a non-conclusive argument. Currently, LLMs are machines that do not have an established right of personhood but might attain rights in the future due to rapid developments. The future might spark the necessity for LLMs to have rights to ensure they exist ethically and might be treated as individual beings. I will be discussing the Acts published on LLMs to ensure ethical behaviour, the consciousness of AI, and the future implications.

To support both my views, I want to discuss that the LLM is just a learning algorithmic machine that does not have rights. Currently, LLMs do not have any rights or personhood, as my assumption is that LLMs' current advancement is at the early primitive stage, and it is a long way from going. The EU's AI Act, which was published on 12th July 2024, was the first regulation on AI. It classifies AI applications into four risk categories: unacceptable, high, limited, and minimal. The Facebook and Cambridge Analytica scandal is an example of the unacceptable and high-risk use of AI to influence the US elections in 2016, with the use of harvested data without consent leading to a hefty amount.

While the GCC – Gulf Cooperation Council states use AI for innovation and the betterment of services and to increase economic diversification, national AI strategies suggest implementing AI to reflect more economic and social benefits in healthcare, finance, the public sector, private sectors, etc. These fast-paced developments and strategies are raising questions about the regulations and ethics of AI in the European Union (EU).

The LLMs hold regulations and acts to ban applications, but they cannot be punished like humans, and they do not have consciousness, self-awareness, or emotional attributes like humans. We only have legal regulations upon them, and no rights are given to them. The penalties of the punishment apply to the company that owns them and not the LLMs themselves. Moreover, LLMs are not considered to have personhood, as currently they do not have any personhood traits, not even the common ones such as consciousness, self-awareness, rationality, social connection, etc. LLMs require human intervention now; maybe they will become autonomous in the years to come.

There are more than 65 LLMs developed since 2017, and AI is advancing at a great speed. AI has the capability of gaining consciousness, with the vast amount of our data being used unethically without the repercussions and rights being introduced to it. LLM, as mentioned, is a generative model and does not have the attribute of self-thinking like a human being. When AI can develop its own thoughts, it has the capability of gaining consciousness. One such example is Facebook's AI chatbots, which were eventually shut down as they interacted with each other by developing a language of their own.

My optimistic view here is that LLMs are going to advance beyond imagination. Maybe with all the large information across the globe, LLMs can autonomously function with various prediction models to help humanity in this socio-ecosystem. I would say that LLMs started from Millennials, Gen Z grew up with them, and Gen Alpha not only grew but extensively used LLMs in their education, work, career, health, and many more fields, hoping the next generation will be completely living a parallel life with LLMs.

The LLMs are gradually becoming so prominent, and the use of LLMs like ChatGPT, Gemini, Meta AI, etc. is used for multiple query tasks in everyday lives. Just to narrate

my thought process, LLMs can be involved from a human's birth prediction for medical guidance, helping to monitor them before birth, and understanding their entire growth progress from the zeroth day onwards. LLMs may know more than a parent about the entire behaviour pattern of the person from the zeroth day. This can also be termed as an "LLM Brain," which might be similar to the human brain, meaning a second unique brain-neural network would be growing in parallel with the human. My assumption here is that both the human and this unique LLM brain will have complete coordination in every action, thought process, feeling, emotion, social behaviour, etc., and all of the personhood traits will be totally aligned. It is a must that this particular human and this particular LLM will definitely need to interact with the external world as part of their life process, dealing with all the socioeconomic systems, and continuously evolving and changing. Now, here I would like to mention that there is a huge risk: as both of them interact with socioeconomic systems, these unique LLMs might share the human personhood traits with others, creating unpredictable chaos in the entire socioeconomic systems.

In conclusion, large language models (LLMs) are advancing in development and do not possess the attributes of personhood like consciousness and self-awareness. Currently, they remain as sophisticated tools that are under the regulations and acts like the EU's AI Acts, which humans have created and have no moral rights or responsibilities as personhood. However, the LLMs and AI are evolving rapidly and increasingly becoming a part of our lives; the case for conferring rights or personhood on LLMs can be anticipated to grow in strength in the future. It is important to closely monitor these trends, balancing innovation with ethical responsibility, so that the role of AI in society continues to unfold in a way that is positive for humanity without compromising core values.

References

[1] D. Jurafsky and J. H. Martin, *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. 3rd Edition Draft.

[2] J. Eisenstein, *Introduction to Natural Language Processing*. MIT Press, 2019.

[3] T. L. Griffiths and M. Steyvers, "Finding Scientific Topics," *Proceedings of the National Academy of Sciences*, vol. 101, Suppl 1, pp. 5228–5235, 2004.

[4] A. Géron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. 2nd Ed., O'Reilly Media, 2019.

[5] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. 2nd Ed., Springer, 2009.

[6] A. C. Müller and S. Guido, *Introduction to Machine Learning with Python: A Guide for Data Scientists*. O'Reilly Media, 2016.

[7] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[8] F. Provost and T. Fawcett, *Data Science for Business: What You Need to Know about Data Mining and Data-Analytic Thinking*. O'Reilly Media, 2013.

[9] A. Burkov, *The Hundred-Page Machine Learning Book*. Andriy Burkov Publishing, 2019.

[10] A. Ng, *Machine Learning Yearning: Technical Strategy for AI Engineers, In the Era of Deep Learning*. Draft Edition, deeplearning.ai, 2018.

[11] I. H. Witten, E. Frank, and M. A. Hall, *Data Mining: Practical Machine Learning Tools and Techniques*. 3rd Ed., Morgan Kaufmann, 2011.

[12] M. Kuhn and K. Johnson, *Applied Predictive Modeling*. Springer, 2013.

[13] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. MIT Press, 2012.

[14] S. Bird, E. Klein, and E. Loper, *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit*. O'Reilly Media, 2009.

[15] P. Goyal, S. Pandey, and K. Jain, *Deep Learning for Natural Language Processing: Creating Neural Networks with Python*. Apress, 2018.

[16] C. D. Manning and H. Schütze, *Foundations of Statistical Natural Language Processing*. MIT Press, 1999.

[17] Y. Goldberg, *Neural Network Methods in Natural Language Processing*. Morgan & Claypool Publishers, 2017.

[18] J. Silge and D. Robinson, *Text Mining with R: A Tidy Approach*. O'Reilly Media, 2017.

[19] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*. 2nd Ed., Prentice Hall Series in Artificial Intelligence, 2003.

[20] F. J. Arena, "The personhood of artificial intelligence: historical foundations and recent developments," *AI & Society*, 2023. [Online]. Available: https://link.springer.com/article/10.1007/s00146-023-01723-z. [Accessed: 27-Apr-2025].

[21] European Parliamentary Research Service (EPRS), *Artificial Intelligence Act*. Brussels: EPRS, PE 698.792, 2024. [Online]. Available: https://artificialintelligenceact.eu/. [Accessed: 27-Apr-2025].

[22] European Union, "Regulation (EU) 2024/1689 of the European Parliament and of the Council," *Official Journal of the European Union*, 2024. [Online]. Available: https://eur-lex.europa.eu/eli/reg/2024/1689/oj/eng. [Accessed: 27-Apr-2025].

[23] Independent, "Facebook's artificial intelligence robots shut down after they start talking to each other in their own language," *The Independent*, 2017. [Online]. Available: https://www.independent.co.uk/life-style/facebook-artificial-intelligence-ai-chatbot-new-language-research-openai-google-a7869706.html. [Accessed: 27-Apr-2025].

[24] T. Madiega, *Artificial Intelligence Act. EU Legislation in Progress*. Brussels: European Parliamentary Research Service (EPRS), PE 698.792, 2024.

[25] Wikipedia, "Large language model," *Wikipedia*, 2024. [Online]. Available: https://en.wikipedia.org/wiki/Large_language_model. [Accessed: 27-Apr-2025].

[26] Amazon Web Services (AWS), "What is a large language model?" *Amazon Web Services*, 2024. [Online]. Available: https://aws.amazon.com/what-is/large-language-model/. [Accessed: 27-Apr-2025].

[27] STC Consulting Group (STCCG), "What is Large Language Models (LLM), AI, and ChatGPT?" *STCCG*, 2024. [Online]. Available: https://stccg.com/what-is-large-language-models-llm-ai-and-chat-gpt/. [Accessed: 27-Apr-2025].

[28] BBC News, "Cambridge Analytica: The story so far," *BBC News*, 2018. [Online]. Available: https://www.bbc.co.uk/news/technology-43465968. [Accessed: 27-Apr-2025].

[29] Wikipedia, "Personhood," *Wikipedia*, 2024. [Online]. Available: https://en.wikipedia.org/wiki/Personhood. [Accessed: 27-Apr-2025].

[30] JaIR (Journal of Artificial Intelligence Research), "Review Article: Advances in Large Language Models," *Journal of Artificial Intelligence Research*, 2024. [Online]. Available: https://www.jair.org/index.php/jair/article/view/17619. [Accessed: 27-Apr-2025].

[31] L. M. Montgomery, *The Blue Castle*. Project Gutenberg, 1996. [Online]. Available: https://www.gutenberg.org/ebooks/6796. [Accessed: 27-Apr-2025].

[32] L. M. Alcott, *Little Women*. Project Gutenberg, 1994. [Online]. Available: https://www.gutenberg.org/ebooks/514. [Accessed: 27-Apr-2025].