Applied Machine Learning Report

Word Count: 2,750

0. Executive Summary

Task 1 - Summary

Task 1 was completed to deliver a machine learning model (ML) that predicts categories based on feature input; the algorithm utilised multi-layer perceptron (MLPClassifier). According to the client's requirements, the model can be considered moderately successful as it realises an overall accuracy score of 91% which is significantly high compared to a bassline approach like random guessing which expects 20%. From analysis of hyperparameters, test scores, precision and recall, the results suggest the model did not overfit. However, it failed to completely meet the misclassification limit which suggests the model's lack of ability to critically distinguish between classes.

Task 2 - Summary

Task 2 was completed to deliver a prototype of a predictive ML model and required the labelling of 100 subset data points (minimum) to equip the algorithm with data necessary to train the model; the algorithm utilised the random forest algorithm (RandomForestClassifier). This prototype can be considered eligible for further development by the client as the overall accuracy is 89% which is a comparatively higher than guessing the majority class all the time (50%). Additionally, the metrics attained from test scores and the classification report suggest the model does not overfit.

1. Data exploration and assessment

The dataset considers several features related to the content posted on NotTheRealWikipedia and suggests that users can upload text (paragraphs) on their website to inform a system that classifies content by context of the data and users that have interacted with it. The company has requested the creation of machine learning algorithms that predict target values based on the labelled features provided. The project consists of two tasks: ‘using machine learning (ML) to identify the category of a paragraph for specific topics of interest, and the creation of a prototype to inform an algorithm that predicts the clarity of text.’

A key component of the dataset is ‘has_entity,’ a categorical feature containing long string values that specify a paragraph’s reference to an organisation, product or person. The column ‘category’ is also a categorical variable that represents the classification of paragraphs based on their context which is an important target of the machine learning model as it provides a task purpose in ensuring connections are made to the desired output of the model.

The distribution of text across classes is relatively imbalanced as the dataset consists of paragraphs mostly related to ‘biographies, compared to ‘movies about artificial intelligence,’ the least designated category.

category	**count**
biographies	2865
philosophy	2465
programming	1924
artificial intelligence	1509
movies about artificial intelligence	161
Philosophy	13
Biographies	13
Programming	10
Artificial intelligence	10
Movies about artificial intelligence	1

Table 1: Original *category* column *(after removal of duplicates)*

The difference between the prevalence of classes highlights the imbalance of data which could prove challenging in model training as ML relies on learning patterns that can be better inferred from 'common' features and because of scarcity, the model may be inclined to inaccurately classify classes of a lower frequency due to the relative lack of data points. This suggests the need for oversampling in preprocessing, as this technique makes up for the lack of minority classes.

As 'paragraph' is a significant feature of the dataset, analysis of the length of text can build an understanding of data complexity levels and highlight the prominence of a paragraph. This additional feature has been derived from clean text to inform the model in formulating learning patterns. As concluded in Table 2, the skewed distribution favours short form text (mean relatively close to the minimum text length) and indicates the possibility of shorter outliers that may influence the overall accuracy of the data, both relevant factors to consider in model building.

Most attributes within the dataset have been labelled prior to delivery, though the company later requires the entry of 100 data points (minimum) that specify the text clarity of respective paragraphs.

text_length – *important values*	**count**
highest	3983
mean	516.67 (2 d.p.)
Lowest	2

Table 2: text_length column *(showcasing highest, lowest and mean values)*

has_entity	**count**
ORG_YES_PRODUCT_NO_PERSON_YES_	2957
ORG_NO_PRODUCT_NO_PERSON_NO_	2739
ORG_YES_PRODUCT_NO_PERSON_NO_	1427
ORG_NO_PRODUCT_NO_PERSON_YES_	1332
ORG_YES_PRODUCT_YES_PERSON_YES_	291
ORG_YES_PRODUCT_YES_PERSON_NO_	122
ORG_NO_PRODUCT_YES_PERSON_YES_	63
ORG_NO_PRODUCT_YES_PERSON_NO_	42
data missing	24

Table 3: Original *has_entity* column *(after removal of duplicates)*

2. Data preprocessing

‘Data consistency refers to the integrity and validity of data that represent real-world entities, and integrity constraints tend to affect the accuracy of the model.’ (Al-Janabi and Janicki, 2016, p.492) Data cleaning, embedding and encoding are collectively aspects that make up preprocessing and occur prior to splitting, to encourage consistency and limit constraints such as bias and missing values, to ensure a fair evaluation of the model.

The company data was allocated to represent feature (paragraph, has_entity) and target (category) values; X and y, which facilitates the preparation of the ML model.

The cleaning process involves the removal of 350 duplicate rows across the dataset, 24 string values indicating ‘data missing’ within has_entity and 61 null values within ‘category’ (as seen in Table 1 and 3) to reduce inconsistencies and improve the efficiency of data processing. An imputation technique, SimpleImputer was used to replace the missing data within ‘has_entity’ and ‘category’ with their respective most frequent values.

‘Mean imputation preserves the mean of the observed data but can be considered less accurate than other impute techniques’ (Chehal et al., 2023). Nevertheless, the application of this method counters the missing data, so the assumption of the most frequent value acts as a ‘filler’ to preserve the distribution of data. Additionally, the ‘category’ column underwent lowercasing to amend 10 uppercased string values and group the labels accordingly to ensure consistency in data quality.

Python’s clean-text package was implemented to return a cleaned version of text from the paragraph column, free of irregularities and noise that could negatively affect the data.

Task 2 adopted a similar approach to cleaning the prototype, applicable to the selected features.

The next stage consists of word embedding, ‘the process of converting words into numbers that computers can process since they cannot understand natural language as is’ (Kim and Jeong, 2021, p.99954). This process utilises a medium natural language processing model from ‘spaCy,’ over a smaller and larger model to establish balance between efficiency and performance. The chosen model holds a relatively great library of vocabulary useful for ensuring accuracy and reducing levels of misclassification. This process considers clean text as an input and generates a vector of 300 dimensions (within a newly merged data frame), which equates to the number of column splits and suggests the presence of 300 unique features related to the document that can help to inform the model. Task 2 requires the small model which offers 96 splits, to cater the size of the prototype.

3. Data encoding and splitting

‘Data encoding is essential for the assignment of numerical values to the selected categorical variables (in this case integers and binary values) to ensure suitability for machine learning models’ (Qiu and Liu, 2023, p.448). OneHotEncoder derives multiple columns of a binary nature that represent each possibility of ‘has_entity’ as a column to retain a recognisable structure, resulting in better interpretable data.

Similarly, the categorical output variables consist of five ‘target’ labels and employs LabelEncoder as a simple approach to expressing each category as an integer, also easily interpreted by a machine learning model for improved performance.

The data assessment highlighted the disparity in the text length data which suggests a requisite for scaling. ‘StandardScaler' is a scikit-learn tool that is effective for standardising features by removing the mean and scaling to unit variance’ (Scikit-learn, 2024). The benefit of this is the standardisation of distribution which calculates a new range that ensures better proportion in text length to equitably contribute towards the model’s learning. In Task 2, scaling caters to all the input features used as they are of relatively similar ranges so collectively, the values can help to identify a common variance useful for advanced analysis.

‘Synthetic Minority Oversampling Technique (SMOTE) is a statistical technique that generates new instances from existing minority cases supplied as input’ (Microsoft, 2024). As SMOTE requires numerical data to identify minority classes and create synthetic samples, the encoded dataset can benefit from an achieved balance and more learning patterns established in relation to minority of data.

4. Data splitting

Data splitting requires the input of optimised feature and target values and utilised a test size of 0.2 to specify the ratio of training and evaluation data (80% and 20% respectively). Task 2 uses a slightly higher test size of 0.25 to make up for the smaller size of data. These proportions allow for a better understanding of correlation in the learning process as the model employs more data to form patterns, but at the expense of a less accurate generalisation performance, which is logical considering the skewed nature of the dataset.

The process randomises the chosen dataset before splitting to avoid bias from ordered data points and trains data to perform well with unseen data as it learns from generalisation, limiting the effect of the implemented test size. To complement this process, a random state applied with a seed value of 42, encourages reproducibility in the 'random' shuffling process to provide consistency in the generation of training and test data sets.

Lastly, stratify is equated to the output to replicate the proportion of target values in the original data with the training and test data within the split. This also addresses the imbalance of data and ensures the consideration of minority classes in the evaluation of the model's performance.

5. Task 1: Topic classification

5a. Model building

The use of MLPClassifier was deemed appropriate for this task as the technique caters to predicting the classification of categories and works well with non-linear data.

The two hyperparameters used, complementary to this model are defined in the table below:

parameters	values	purpose
clf__hidden_layer_sizes	(30, 40)	The number of hidden layers, their respective neurons and the connections ade that support the model's ability to learn the training data.
clf__alpha	(0.0001, 0.0005)	Enforces L2 generalisation to limit overfitting by controlling the weights that fluence the activity of neurons forming connections.

Table 8: Description of Multi-layer Perceptron parameters

The process of identifying optimal hidden layer sizes requires experimenting with values that cater to the complexity required of this model. This is essential for avoiding overfitting but at the expense of a high processing time in training the model. Alpha parameters can supplement imbalanced data that requires more connections to be made through generalisation to better the chances of predicting unseen data. Simultaneously, experimentation helped to understand the threshold of weights that the model can bear to ensure the right balance for regularisation. Considering oversampling has previously been applied, the alpha can leverage low parameters to ensure the model doesn't overgeneralise and forget the nature of distribution.

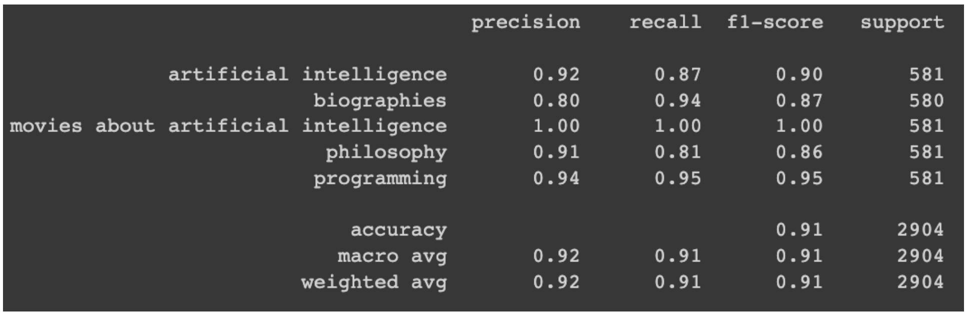
5b. Model evaluation

From the classification report in Figure 1, we can infer that the overall accuracy of the model is 91% which is significantly higher than a random guess trivial baseline (20%). To complement this, the precision and recall figures suggest reliability in the generalisation of the model as the values are competitively high and within a similar range across classes which is a tribute to the model's general success of highlighting the correct positive predictions and identifying the correct true positives.

Test Scores			
indices	**std value**	**mean values**	**correlation**
0	0.002182	0.922951	A commonality can be inferred rom the relationship between low tandard deviations and relatively igh mean values which suggests he model is learning the essential atterns of data and avoiding verfitting.
1	0.001971	0.921057	
2	0.002847	0.925189	
3	0.007290	0.918647	

Table 4: Results of GridSearchCV output – Task 1

GridSearch is a technique that enables the algorithm to find optimal parameters from the specified values. This model uses 5-fold cross validation and accuracy to inform the final scores seen in Figure 1. This also reduces overfitting through training and evaluation of the data.



```
                                  precision    recall  f1-score   support

          artificial intelligence      0.92      0.87      0.90       581
                      biographies      0.80      0.94      0.87       580
movies about artificial intelligence      1.00      1.00      1.00       581
                       philosophy      0.91      0.81      0.86       581
                      programming      0.94      0.95      0.95       581

                         accuracy                          0.91      2904
                        macro avg      0.92      0.91      0.91      2904
                     weighted avg      0.92      0.91      0.91      2904
```

Figure 1: Classification Report 1 for NotTheRealWikipedia

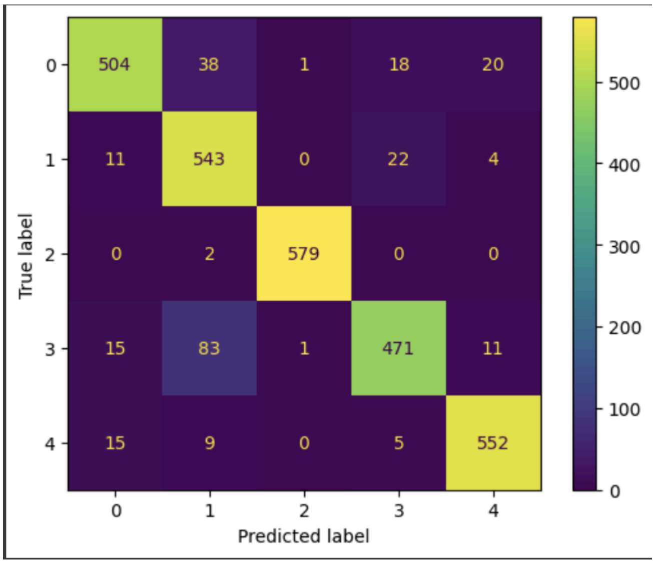


Figure 2: Confusion Matrix 1 for NotTheRealWikipedia

5c. Task 1 Conclusions

- The client requirements state, a successful model should perform better than a trivial baseline, shouldn't be overfit to the training dataset and have no more than 10% of paragraphs get misclassified into an unrelated class. Only 2 of the 3 criteria have been met which makes the model unsuccessful.

- Nevertheless, I would recommend the client keep track of the F1 score as this collectively considers the imbalanced distribution of the original dataset and displays the model's ability to recognise true positives and minimise false instances.

6. Task 2: text clarity classification prototype

6a. Ethical discussion

There are ethical implications of automatically rejecting users' work based on predicted text clarity. As automation lacks human essence in deciphering text, the performance of the model may be limited in understanding complex entries of data or intricacies of language use which could negatively impact users producing content of this nature. An approach to supporting this model could include error messages that give user's feedback on the rejection of the text and ways to abide by the guidelines.

A certain category that may be overrepresented within the training data could skew the distribution of information and create an automated bias in the learning patterns formed by the machine learning model which could hinder its performance as the model doesn't know any better to produce coherent results. As a result, the misclassification rate may increase as primary learning is predominantly made up of the majority class. Relative to the algorithm, the majority class is 'clear_enough' which suggests the model could negatively affect the reputation of the company if 'not_clear_enough' text is mistakenly accepted, with potential for viewers to encounter explicit or irrelevant content that could consequently affect the standard and sustainability of the company.

6b. Data labelling

In labelling the subset used in the prototype, I adopted a manual approach which entailed the use of discretion to decipher the paragraphs and allocate the text clarity label accordingly; having an understanding for the definition of both label supported the process: 'clear_enough' means the paragraph depicts its respective category and 'not_clear_enough' means the paragraph doesn't portray the topic in the category.

Aside from the previous denotation, the comprehension of topics within the category can encourage initiative in decision making. For instance, biographies tend to be written in third person, therefore looking for references to other people or the use of pronouns can be an indication; in artificial intelligence, any reference to machine learning techniques, neural networks, natural language processing or generally any AI techniques is sufficient to qualify.

I am highly confident of this manual approach to data labelling as humans can fundamentally utilise common sense to decipher the text clarity. As mentioned in the ethical discussion, humans are likely to better grasp the intricacies of complex data as opposed to an algorithm, which leverages reasoning power in the manual allocation of labels.

text_clarity	**count**
clear_enough	94
not_clear_enough	30

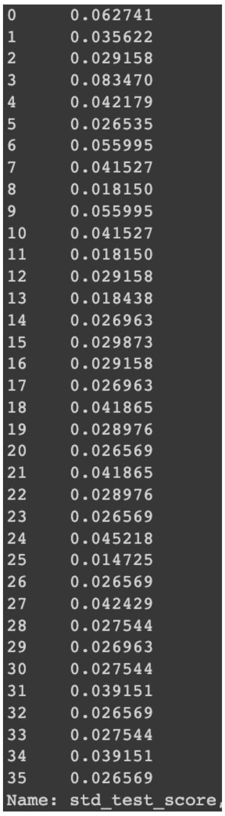
Table 5: text_clarity column labels *(124 data points used)*

paragraph	text_clarity	reasoning
The latest revision of the language was earlier referred to as ortran 2015. It is a significant revision and was released on ovember 28, 2018. Fortran 2018 incorporates two previously ublished Technical Specifications.	clear_enough	As Fortran is a rogramming language, his identifies with the *rogramming'* category llocated which suggests he text clarity is clear nough.
Book II is devoted to topics relating to arguments where an accident" is predicated of a subject.	not_clear_enough	As there isn't a person eferenced in the text, this ails to identify with biographies' as a category hich suggests the text's larity is not clear.

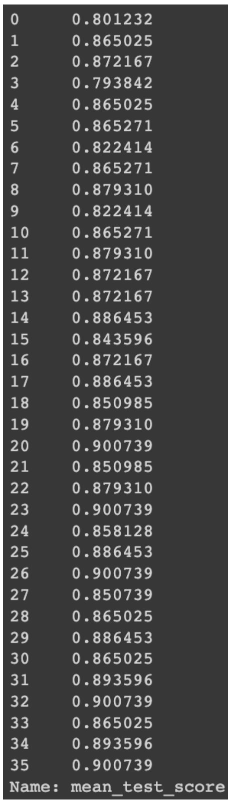
Table 6: Examples of the paragraph that identifies as clear_enough and not_clear_enough

6c. Model building and evaluation

The use of dimensionality reduction (PCA) in this task, is relevant to the entirety of the model as the processed data produces multi-dimensional vectors and considering the numerical data is derived from a relatively small dataset, this suggests a need for reduction in dimensionality to prevent unnecessary noise and preserve the information.



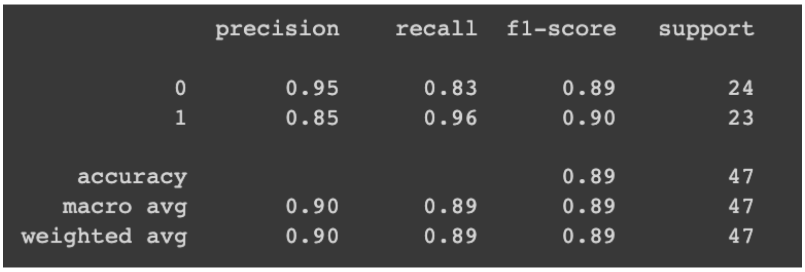
```
0     0.062741
1     0.035622
2     0.029158
3     0.083470
4     0.042179
5     0.026535
6     0.055995
7     0.041527
8     0.018150
9     0.055995
10    0.041527
11    0.018150
12    0.029158
13    0.018438
14    0.026963
15    0.029873
16    0.029158
17    0.026963
18    0.041865
19    0.028976
20    0.026569
21    0.041865
22    0.028976
23    0.026569
24    0.045218
25    0.014725
26    0.026569
27    0.042429
28    0.027544
29    0.026963
30    0.027544
31    0.039151
32    0.026569
33    0.027544
34    0.039151
35    0.026569
Name: std_test_score,
```



```
0     0.801232
1     0.865025
2     0.872167
3     0.793842
4     0.865025
5     0.865271
6     0.822414
7     0.865271
8     0.879310
9     0.822414
10    0.865271
11    0.879310
12    0.872167
13    0.872167
14    0.886453
15    0.843596
16    0.872167
17    0.886453
18    0.850985
19    0.879310
20    0.900739
21    0.850985
22    0.879310
23    0.900739
24    0.858128
25    0.886453
26    0.900739
27    0.850739
28    0.865025
29    0.886453
30    0.865025
31    0.893596
32    0.900739
33    0.865025
34    0.893596
35    0.900739
Name: mean_test_score
```

Figure 3: Results of GridSearchCV output – Task 2

Like Task 1, Figure 3 indicates a positive correlation in the model's ability to learn patterns because of low standard deviation and a relatively high mean. This shows signs that the model does not overfit as the metrics are in favour of generalisation essential to labelling unseen data effectively.



```
              precision    recall  f1-score   support

           0       0.95      0.83      0.89        24
           1       0.85      0.96      0.90        23

    accuracy                           0.89        47
   macro avg       0.90      0.89      0.89        47
weighted avg       0.90      0.89      0.89        47
```

Figure 4: Classification Report 2 for NotTheRealWikipedia

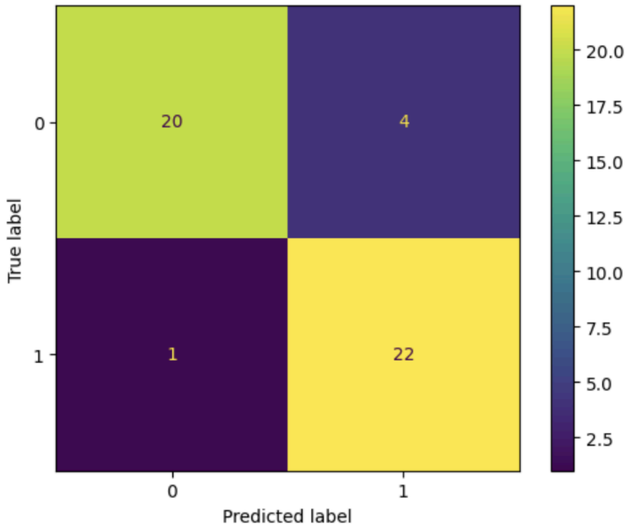


Figure 5: Confusion Matrix 2 for NotTheRealWikipedia

The accuracy as seen in Figure 4, equates to 89% which is a great standard for predicting target variables correctly. Noticeably, the precision for ‘not_clear_enough’ (class 0) is 1, the highest value, which suggests that even though this is the minority class, the model has prioritised the understanding of true positives. Simultaneously, the recall and precision are of high values in the validation stage and is consistent with the results seen in the training data which suggests a model of great performing model.

Though the recall for ‘not_clear_enough’ is reasonably high, the natural bias of a small dataset suggests the importance of focusing on recall to better prevent misclassification of the minority class. However, as seen in the confusion matrix in Figure 5, the misclassification rate for not_clear_enough (Class 1) is 4% which is relatively insignificant.

The Random Forest Classifier model is a suitable choice for the dataset as it reduces the risk of overfitting by leveraging the result of combined predictions to acquire a total that can be used to train the model. The validation process tunes the hyperparameters to guide the performance of the classifier:

parameters	**alues**	urpose
n_estimator	[100, 150, 200]	The estimator represents the number of decision trees used by the iodel and implements the 3 values listed to offer a flexible approach in apturing data and allow for an ideal estimation to be deduced.
max_depth	[3, 5, 10]	The maximum depth of a decision tree; the deeper a tree, the more onnections that can be established during the learning process of aining. The numbers used signify a balance between shallow and ioderate depth which helps to identify an ideal level of depth for the andom Forest.
in_samples_split	2, 5]	The minimum number of samples that the model requires to split a ode. As the threshold is relatively low, this ensures a greater omplexity of trees with more splits due to the low parameters used hich in turn better understands the complexity of relationships.
in_samples_leaf	2, 4]	The minimum samples assigned to finalise the values for lassification.

Table 8: Description of Random Forest parameters

Collectively, the parameters used cater to a smaller dataset and prioritise a reduction of overfitting to ensure an intricate analysis of relevant data points that encourage an accurate representation of the data. Through experimentation, a balance between over and underfitting was achieved to enable a favourable capture of data points relevant to training the model.

6d. Task 2 Conclusions

- According to the client's requirements of a model that doesn’t overfit and has an overall accuracy above 50%, the model can be deemed successful.
- As the dataset is small and imbalanced, keep track of the F1 score to offer a balanced measure of performance.
- A suggestion for improvement would be to include more data points in the subset, allowing the model to better learn patterns and exclude noise.

7. Self reflection

After some reflection, it’s clear the data used to train the Task 1 model can benefit from improvements that prioritise recall, to influence the accuracy of correct positive predictions (particularly class 3); additional techniques such as cost-sensitive learning can be used during the training stage to impose the significance of wrongful classification.

8. References

Al-janabi, S. and Janicki, R. (2016) A Density-based Data Cleaning Approach for Deduplication with Data Consistency and Accuracy, *2016 SAI Computing Conference (SAI)*, pp. 492

Chehal, D., Gupta, P., Gulati, P., Gupta, T. (2023) Comparative Study of Missing Value Imputation Techniques on ECommerce Product Ratings, Informatica, pp. 376

Jihye, K. and Jeong, O. (2021). Mirroring Vector Space Embedding for New Words, 4(7), pp. 99954

Qiu, Q. and Liu, H. 'Numerical Embedding of Categorical Features in Tabular Data: A Survey,' 2023 International Conference on Machine Learning and Cybernetics (ICMLC), pp. 448

Scikit-learn (2024) sklearn.preprocessing.StandardScaler. Available at: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html (Accessed: 23rd March 2024).

Microsoft (2024) SMOTE. Available at: https://learn.microsoft.com/en-us/azure/machine-learning/component-reference/smote?view=azureml-api-2 (Accessed: 23th March 2024).