Implementing Classification Processing Techniques For Tabular Dataset and Sentiment Analysis

Aulia Amirullah Zulkarneidi
Masters of Data Science (with Specialization in Artificial Intelligence)
School of Computing
Newcastle University
Newcastle Upon Tyne, United Kingdom
c2060634@newcastle.ac.uk

***Abstract*—We experiment with possible classifiers to deal with tabular dataset called fars dataset (Fatality Analysis Reporting System) and US Airline Sentiment Dataset. Both datasets are labelled. They can be processed using classification techniques. We make comparisons using Random Forest, Logistic Regression and try using TPOT to find the best classifier to deal with fars dataset which turns out to be K-Nearest Neighbor (KNN) with 0.68 percent f1 accuracy score. We try comparing non-neural network architectures, and two different models using LSTM which is a type of RNN which turns out that has 128 dimensional vectors perform better than that of 256 and the way LSTM regards the sentences as a meaning is perfect for text classification.**

Keywords—FARS, Sentiment Analysis, Random Forest, Logistic Regression, TPOT Classifier, RNN, LSTM.

I. INTRODUCTION

In this analysis, I perform two experiments such as tabular dataset which is called FARS and a US Sentiment Dataset. FARS is a nationwide census providing by NHTSA, congress and the American public annual data about fatal injuries suffered in motor vehicle traffic crashes. This is a tabular dataset which has multiple labels which can be processed using Random Forests [1][2][3].

Random Forests is an ensemble learning algorithm which makes predictions based on an average of their predictions [2]. It is a combination of tree predictors such that each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest [4].

Apart from Random Forests, we also try to experiment with Logistic Regression. Multinomial logistic regressions can estimate the probabilities of many possible outcomes [5]. It is often used for classification and predictive analysis. A multinomial logistic regression model can help to determine the influence of dependent variables to the outcomes. The measurement is by calculating the relationship between a dependent variable and one or more independent variables.

As a comparison, we also implement TPOT Classifier. TPOT pipeline is built to test several different combinations of models, feature preprocessing/ selection methods [6]. TPOT will implement preprocessing techniques such as normalization and the robust scaler, feature reduction such as PCA, helps address concerns models overfit by reducing unimportant features.

For US Airline Sentiment Dataset, we try to implement Random Forests, SVM, Naive Bayes, Decision Tree and implement Bagging to improve the accuracy of Decision Tree and LSTM. Support Vector Machine is one of the supervised learning classification techniques, which is widely applied in several areas such as handwritten digit recognition, face recognition and text categorization [7]. SVM works by finding the hyperplane in an N-dimensional space (N – the number of features) that distinctly classifies the data points. SVM will draw the negative and positive hyperplane which we can also calculate the distance as margin between the two hyperplanes. That's how we classify the data using SVM.

Naïve Bayes classification implements Bayes' Theorem with strong independence assumption between the features. Naïve Bayes classification produces good results for textual data analysis such as Natural Language Processing. It is learned as a generative model that fits the distribution of the document instances given a class label and gives a strong assumption that the term features within a document are conditionally independent given a class label [8].

A decision tree is also used in this experiment regardless of the performance. The concept is used in random forest. It is a map of the possible outcomes which has a node or nodes and branches to make a decision of a given problem. It has a hierarchical, tree structure which consists of a root node, branches, internal nodes and leave nodes.

Since the accuracy of decision tree was actually quite good, but I also did some experiments with Bagging. Bagging (bootstrap aggregation) is the ensemble learning method to reduce variance within a noisy dataset. A random sample of data in training set is selected with replacement which means that individual data can be chosen more than once (repeatedly). These weak models will be trained again independently using the newly generated data samples (subgroups of data which is also known as bags) . It helps improve the performance of weak models [9].

LSTM can solve the problem of the learning approaches storing information over extended time intervals via recurrent backpropagation which takes a very long time [10]. LSTM can learn to bridge minimal time lags and memorize previous state. LSTM is a type of Recurrent Neural Network (RNN) which means that the input can also be the output of the previous steps. It is better than traditional RNNs in terms of memory. It

performs better because it has a good hold over memorizing certain pattern. LSTM has an internal state and has three gates (Forget Gate, Input Gate and Output Gate).

Forget Gate is for the information which can be forgotten as it is no longer relevant. Meanwhile, Input Gate is about what new information should be added or updated into this working storage information and Output Gate is about all the information that's stored in that state, which part of it should be output in this particular instance.

II. FARS Analysis

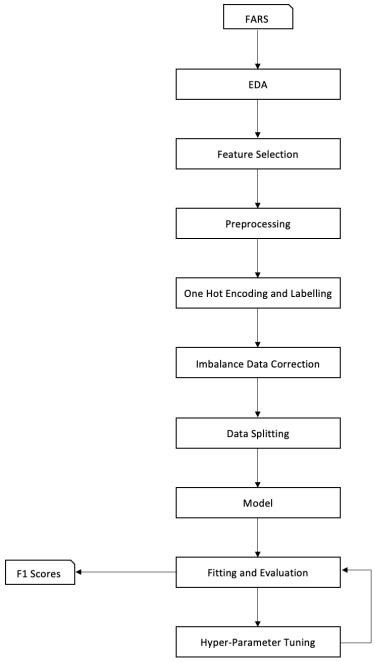

Fig. 1. The Pipeline of FARS Analysis

The first step of the pipeline is EDA (Exploratory Data Analysis) which is meant to find the correlation among variables and between variables and target classes. There are even more things we did such as seeing the distribution of each of the variables and see the mean, min, max of the values of the variables as shown in Fig. 2.

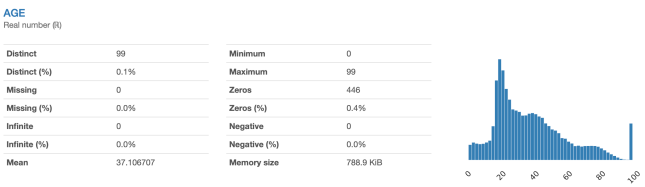

Fig. 2. Age Summary by using EDA Technique

The way I see the multicolinearity between variables and the target classes is by seeing a heatmap which ranges from -1 to 1 in which if the value of the correlation is closer to -1, it means that it has a less strong correlation while a value of 1 means that it has a strong correlation. Correlation heatmaps are important because it helps identify which variables may potentially result in multicollinearity which would compromise the integrity of the model.

After doing EDA, I did missing values checking which turns out that there's no missing value but rather it contains values such as unknown in most of variables as shown in Fig. 3.

Fig. 3. The Pipeline of FARS Analysis

Those are the findings I found during EDA. The next step is feature selection. Removing features that have a high percentage of missing values is a good practice which will improve our accuracy score. It's also hard for Machine Learning Classifiers to learn from datasets which contain too many missing values. I dropped some of the columns because it has too strong correlations with other variables and have weak correlations with our target class which is Injury Severity.

Preprocessing techniques come into play by replacing unknown like words in each of variables with other values available in each of the variables. The next step is implementing One Hot Encoding to sex since sex is a categorical feature. The same is applied to Taken To Hospital variable. The next step is implementing labelling to the other remaining features so that they can be converted into numeric and select all the features as X and injury severity as our y (the target class).

Since we identified there's an imbalance in our targets which will make our model bias as shown in Fig. 4., we need to consider to balance our data. Class 1 seems to oversample the others.

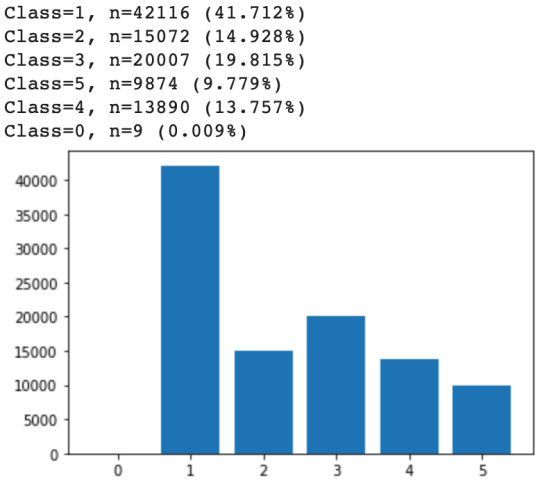

Fig. 4. Imbalance Data

We use a technique to tackle with oversampling which refers to copying or synthesizing new examples of the minority classes so that the number of examples in the minority class better resembles or matches the number of examples in the majority classes. We use a technique called Synthetic Minority

Oversampling Technique (SMOTE) which creates the synthetic examples rather than by over-sampling with replacement. It operates in feature space rather than data space [10].

```
Class=1, n=42116 (16.667%)
Class=2, n=42116 (16.667%)
Class=3, n=42116 (16.667%)
Class=5, n=42116 (16.667%)
Class=4, n=42116 (16.667%)
Class=0, n=42116 (16.667%)
```

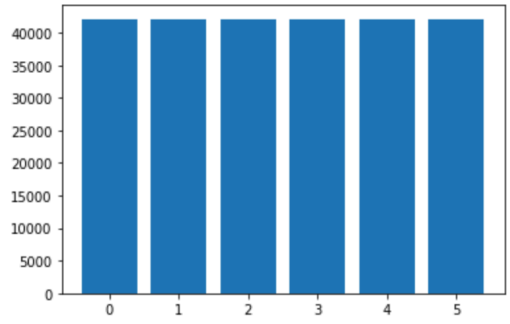

Fig. 5. Balanced Data

After we balance our data, we need to split our data to be the training dataset and test dataset so that our model won't be biased and is used to the whole dataset. The splitting is meant to check whether our model is good enough for some amount of data which were never trained before so that our models can be accurate.

Then, we define three models such as Random Forest, Multinomial Logistic Regression and TPOT. I tried picking n_estimators as 100 and Max_depth None as the first attempt for this particular dataset. The next step is we fit our training data to our model and then we also evaluate our model using our test dataset which turns our as follows:

```
Accuracy =  0.7797491063066391
Precision =  0.7797491063066391
Recall =  0.7797491063066391
f1 =  0.7797491063066391
f1_train =  0.9485038470888194
[[12685     0     0     0     0     0]
 [    2  8652  1435  1250   911   478]
 [    0   923  9011    20  1871   739]
 [    1   720    12 11159    89   719]
 [    0   650  1954   237  8565  1200]
 [    0   456   777  1089  1164  9040]]
```

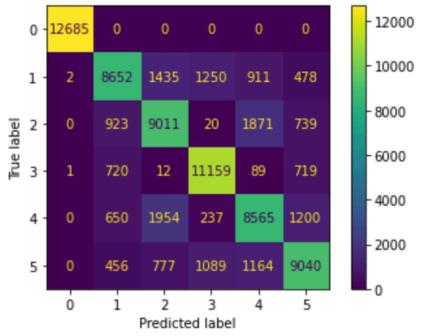

Fig. 6. Random Forest Fitting and Evaluation Result

In Fig. 6, we can see that our score is quite high already which is 0.7797. However, our training f1 score is 0.9485 which means that our training dataset overfits our model a bit and our test dataset is about 0.1688 in difference. Since random forest works by considering the majority of the forests we define saying a particular class (n_estimator as the number of trees) and max_depth is defined as how many nodes are expanded until all leaves are pure.

Therefore, the next step is to tune hyper-parameters. I tried different combinations of n_estimators and max_depth as what Randomized Grid Search Cross Validation suggests, but it turns out that it doesn't improve much. Among 5 times run I used for running Randomized Grid Search Cross Validation, it turns out that n_estimators 200 with max_depth is set to None improves our accuracy to 0.82 which reduces the overfitting a bit as shown in Fig. 7.

```
Accuracy =  0.7812660765872126
Precision =  0.7812660765872126
Recall =  0.7812660765872126
f1 =  0.7812660765872126
f1_train =  0.9485095004155195
[[12685     0     0     0     0     0]
 [    2  8636  1470  1253   908   459]
 [    0   943  9017    17  1888   699]
 [    1   701    11 11179    89   719]
 [    0   644  1919   236  8612  1195]
 [    0   442   735  1091  1160  9098]]
```

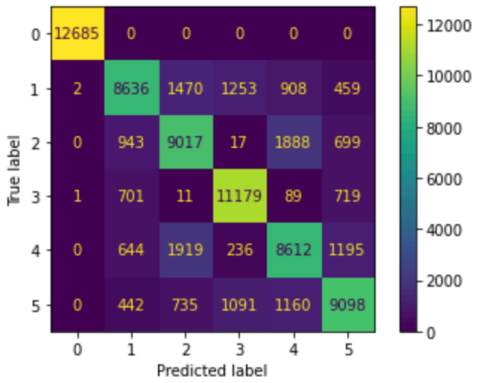

Fig. 7. Cross Validation of Random Forest

Cross Validation is a statistical method to estimate the performance of machine learning models which is commonly used to compare and select a model for a given predictive modelling problem because it results in skill estimates that generally have a lower bias than other methods. We use stratified cross validation which means that the process of splitting data into k-folds by ensuring that each fold has the same proportion of observations with a given categorical value, such as the class outcome value.

Grid search cross validation is used to tune hyper-parameters while the randomized version is selecting the combination of param grid randomly to find the best parameters of our models.

The other model we tried is Multinomial Logistic Regression. In Logistic Regression, it turns out that our accuracy score is even way less that our random forest, even though we've tuned our hyper-parameters many times with different combinations.

```
Accuracy =  0.6078961600865332
Precision =  0.6078961600865332
Recall =  0.6078961600865332
f1 =  0.6078961600865332
[[12475     1     0     0     0     0]
 [  372  6603  1592  1968  1738   367]
 [    5  1012  4696    22  6389   635]
 [  318   829     3 11252    17   106]
 [   53   769  2208   330  8077  1264]
 [  279   806   895  2990  4757  2981]]
```

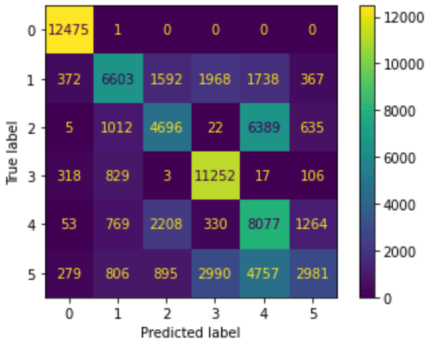

Fig. 8. Cross Validation of Logistic Regression

The next model is to use TPOT pipeline. TPOT will provide our data preparation, scaling, model and hyper-parameters for us and suggests which classifier we should use along with the best parameters we can use. It shows us that KNN with the number of neighbors is equal to 29, power of parameter that we need to use is 1 which is equal to Manhattan Distance and weights is equal to distance (which means that closer neighbors of a query point will have a greater influence than neighbors which are further away.

III. US AIRLINE TWITTER SENTIMENT ANALYSIS

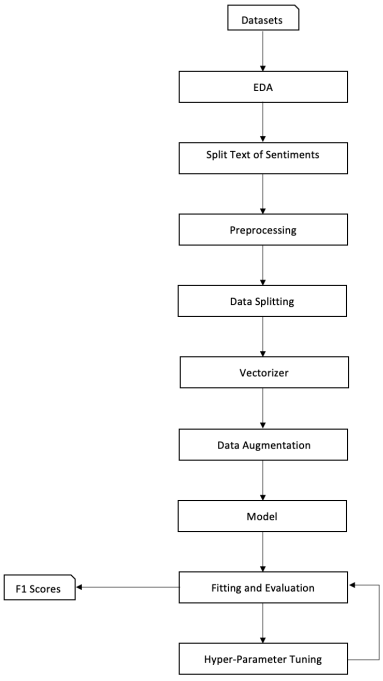

Fig. 9. The Pipeline of US Airline TWITTER Sentiment Analysis

We have three datasets which are provided and are already split between training, validation and test set. In the EDA part, we just explored our datasets starting from variables and the shape of the datasets. It turns out that we have tweet_id, text and airline_sentiment.

```
training_set.tail()
```

	tweet_id	text	airline_sentiment
11853	570123872168574976	@AmericanAir will not help us on the phone, at...	negative
11854	570063683256242177	@USAirways has the worst customer service line...	negative
11855	568032524749942784	@USAirways grades for this trip:\n\nFlight tim...	negative
11856	569705813142409217	@united Thanks for the vague canned response t...	negative
11857	569976114124349440	@united already did that at the airport and 12...	negative

```
validation_set.head()
```

	tweet_id	text	airline_sentiment
0	568107472260624384	@southwestair Great job celebrating #MardiGras...	positive
1	568215698524246016	@southwestair thanks for taking it up a notch!...	positive
2	567842466851905536	@USAirways Being put back on hold for what has...	negative
3	568834824410148864	@united Thank you for your offer! All sorted o...	positive
4	569590527349252096	@JetBlue wondering if it's possible for my col...	neutral

Fig. 10. The shape of datasets

In our training set, we have 11857 sentiments, 1317 sentiments in our validation set and 1463 sentiments in our test set.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 11858 entries, 0 to 11857
Data columns (total 3 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   tweet_id           11858 non-null  int64
 1   text               11858 non-null  object
 2   airline_sentiment  11858 non-null  object
dtypes: int64(1), object(2)
memory usage: 278.0+ KB
```

```
validation_set.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1318 entries, 0 to 1317
Data columns (total 3 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   tweet_id           1318 non-null   int64
 1   text               1318 non-null   object
 2   airline_sentiment  1318 non-null   object
dtypes: int64(1), object(2)
memory usage: 31.0+ KB
```

```
[ ] test_set.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1464 entries, 0 to 1463
Data columns (total 4 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   Unnamed: 0         1464 non-null   int64
 1   tweet_id           1464 non-null   int64
 2   text               1464 non-null   object
```

Fig. 11. The shape of datasets

We also saw the distributions, mean, min, max, standard deviation of the datasets. We also check the NaN values of the datasets so that we knew whether we had to deal with the missing values or not.

We also found out that our class is imbalanced after visualizing our datasets as follows:

```
Total number of sentiments of tweets :
negative    7434
neutral     2510
positive    1914
Name: airline_sentiment, dtype: int64
```

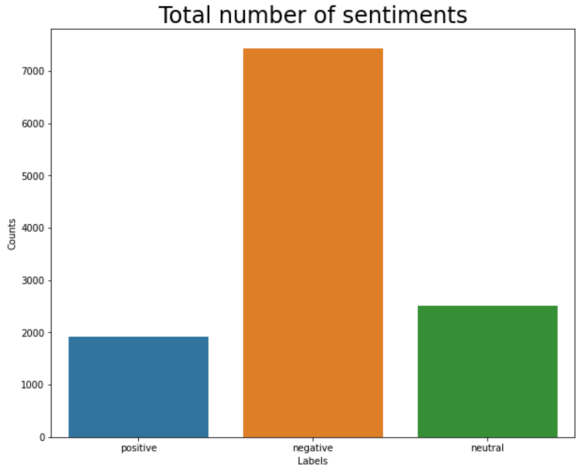

Fig. 12. Imbalanced Data

The next step is preprocessing. In this step, we convert the sentiments which are negative, neutral and positive into numeric. Then, we remove html tags, emojis, url, punctuations, stop words, usernames, unique character and transform contracted words into deconstructed form. It will make the process of transforming words into numeric become easier later.

Then, we split the data into X and y based on the datasets either they are train set, validation set or test set. We actually performed two experiments (with and without data augmentation). Turns out the result of without data augmentation is better. But, without the data augmentation technique, our models might be biased towards the negative sentiment instead.

The next step is to transform our X and y into numeric then process them to balance both of them. After balancing our variable and target class. We can build our random forest, SVM, Naïve Bayes and decision tree.

We then tune our hyper-parameter and optimize them using the best parameters that randomized grid search cross validation suggests. We also implement Bagging to improve the accuracy of Decision Tree.

For Random Forest, we chose n_estimators as 250 as it improves the accuracy 0.004 rather than using n_estimators as 100 for our datasets. In this case, the more forests we have seems the better it will be because we're dealing with texts.

SVM performs better with the regular parameter C as 1.0 and the way of splitting the the data using rbc kernel. After going through the cross validation, turns out that C as 0.5 with linear kernel performs a bit better.

Multinomial Naïve Bayes seems perform better than Random Forest and SVM. With the learning rate suggested by grid search cross validation as 0.5 so that our model doesn't overfit can perform 0.03 better than setting the alpha as 1.0 or the value above.

We have two architectures of LSTM, the first one is with 128 dimensional vector and the other one is with 256 dimensional vector. Here are the architectures:

```
Layer (type)                 Output Shape              Param #
=================================================================
input_2 (InputLayer)         [(None, None)]            0

embedding_1 (Embedding)      (None, None, 128)         1280000

bidirectional_2 (Bidirectio  (None, None, 128)         98816
nal)

bidirectional_3 (Bidirectio  (None, 128)               98816
nal)

dense_1 (Dense)              (None, 1)                 129

=================================================================
Total params: 1,477,761
Trainable params: 1,477,761
Non-trainable params: 0
_________________________________________________________________
```

Fig. 13. 128 Dimensional Vector of LSTM

```
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         [(None, None)]            0

embedding (Embedding)        (None, None, 256)         2560000

bidirectional (Bidirectiona  (None, None, 256)         394240
l)

bidirectional_1 (Bidirectio  (None, 256)               394240
nal)

dense (Dense)                (None, 1)                 257

=================================================================
Total params: 3,348,737
Trainable params: 3,348,737
Non-trainable params: 0
_________________________________________________________________
```

Fig. 14. 256 Dimensional Vector of LSTM

The difference between the two is that the number of the nodes. I did two experiments but turns out if I increased the number of dimensional vector, the performance is getting worse. The best I could get is that 128 dimensional vector as it won't overfit or underfit the trained model.

IV. Experimental Results

It turns out that Random Forest and KNN performs better for our tabular datasets while LSTM and Naïve Bayes outperforms other models and LSTM performs better in terms of memory and the way it treats the sentences based on the maning of the sentences and words. Here are the f1 score accuracy tables:

No.	Model	F1 Score
1.	Random Forest	0.7812
2.	Multinomial Logistic Regression	0.6114
3.	TPOT (KNN)	0.6885

Fig. 15. F1 Score for FARS Dataset Analysis

No.	Model	F1 Score
1.	Random Forest	0.7056
2.	SVM	0.7540
3.	Naïve Bayes	0.7848
4.	Decision Tree	0.6318
5.	Bagging of Decision Tree	0.6523
6.	LSTM (128 Dimensional Vectors)	0.7734
7.	LSTM (256 Dimensional Vectors)	0.7704

Fig. 16. F1 Score for US Airline TWITTER Sentiment Analysis

Multinomial Logistic Regression overfits our training dataset which is something we actually should avoid. Decision tree in our text dataset analysis also doesn't perform quite well even though bagging can improve the performance a little bit. LSTM and Naïve bayes performs better and they are suitable for text classification, and they're widely used for classifying sentiments.

REFERENCES

[1] Lin W., Zhihao M., Heng X., Tingxia M., Cheng H., Jiaxin C., Xiaohu Y.. Status diagnosis and feature tracing of the natural gas pipeline weld based on improved random forest model. International Journal of Pressure Vessels and Piping, Volume 200, 2022, 104821, ISSN 0308-0161. https://doi.org/10.1016/j.ijpvp.2022.104821.

[2] Wei G., Fan X., Zhi-Hua Z. Towards convergence rate analysis of random forests for classification. Artificial Intelligence, Volume 313, 2022, 103788, ISSN 0004-3702, https://doi.org/10.1016/j.artint.2022.103788.

[3] Qiushi S, Minghui H, Ponnuthurai N. S., Rakesh K. Weighting and pruning based ensemble deep random vector functional link network for tabular data classification. Pattern Recognition, Volume 132, 2022, 108879, ISSN 0031-3203. https://doi.org/10.1016/j.patcog.2022.108879.

[4] Breiman L. (2001) Machine Learning, 45 (1), pp. 5 - 32, Cited 64013 times. DOI: 10.1023/A:1010933404324.

[5] Didier N., Trevor J. H., Multiclass-penalized logistic regression, Computational Statistics & Data Analysis, Volume 169, 2022, 107414, ISSN 0167-9473. https://doi.org/10.1016/j.csda.2021.107414.

[6] Pedro M., R.A. B., Leslie K. Expanded analysis of machine learning models for nuclear transient identification using TPOT. Nuclear Engineering and Design, Volume 390, 2022, 111694, ISSN 0029-5493, https://doi.org/10.1016/j.nucengdes.2022.111694.

[7] B. Ramesh, J.G.R. Sathiaseelan. An Advanced Multi Class Instance Selection based Support Vector Machine for Text Classification. Procedia, Computer Science, Volume 57, 2015, Pages 1124-1130, ISSN 1877-0509. https://doi.org/10.1016/j.procs.2015.07.400.

[8] Han-joon Kim, Jiyun Kim, Jinseog Kim, Pureum Lim, Towards perfect text classification with Wikipedia-based semantic Naïve Bayes learning, Neurocomputing, Volume 315, 2018, Pages 128-134, ISSN 0925-2312. https://doi.org/10.1016/j.neucom.2018.07.002.

[9] Giang N., Rodney B., Rohitash C. Evolutionary bagging for ensemble learning, Neurocomputing, Volume 510, 2022, Pages 1-14, ISSN 0925-2312, https://doi.org/10.1016/j.neucom.2022.08.055.

[10] Chawla, N. V. et al. (2002). SMOTE Synthetic Minority Over-Sampling Technique.Journal of Artificial Intelligence Research, 16, 321-357.